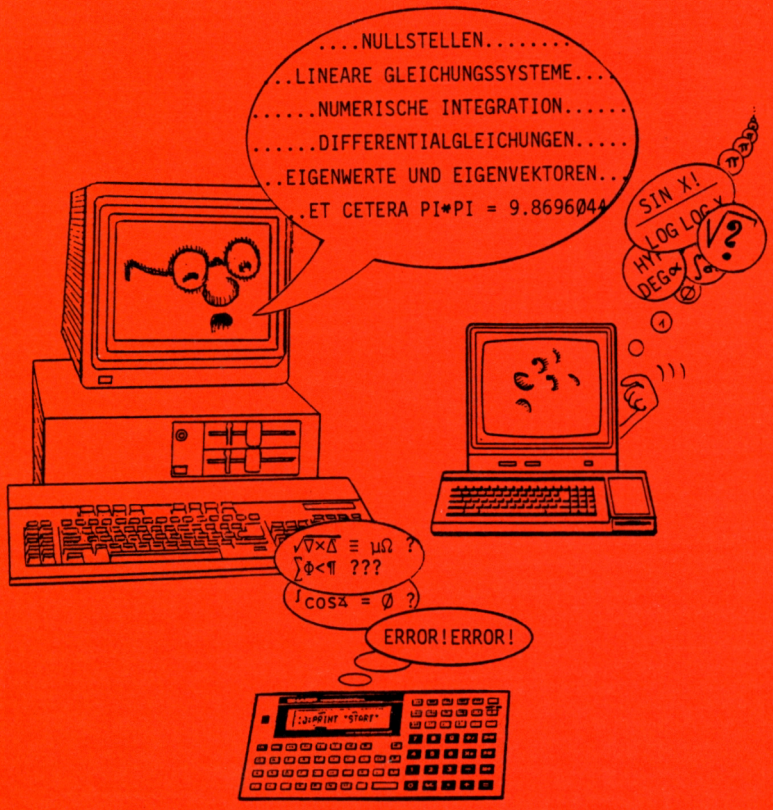
MERZIGER · WIRTH

BASIC PROGRAMME ZUR HÖHEREN MATHEMATIK



ÜBER 60 PROGRAMME MIT ANLEITUNGEN UND BEISPIELEN

BASIC PROGRAMME
ZUR
HÖHEREN MATHEMATIK

Dr. Gerhard Merziger
Dr. Thomas Wirth

3. Auflage
Alle Rechte vorbehalten.
Verlag: **C. Feldmann, Am Bergfelde 28, 3257 Springe 2**
Druck: Linden-Druck GmbH, Fössestr. 97 A, 3000 Hannover 91

Zu beziehen beim Verlag oder im Buchhandel.
ISBN 3-923923-15-5

Hannover, Herbst 1988

Im Buchhandel oder direkt beim Verlag zu erhalten :

Dietrich Feldmann

Repetitorium der Ingenieurmathematik Teil 1

Standardarbeitsbuch zur Ingenieurmathematik. Ausführliche Erklärungen der Rechenverfahren mit über 1000 durchgerechneten Beispielen und Aufgaben.

ISBN 3-923923-00-7 332 Seiten ca. 19 DM

Repetitorium der Ingenieurmathematik Teil 2

Numerische Verfahren für lineare Gleichungssysteme und Eigenwertaufgaben / Variationsrechnung / Ritz-Verfahren / Laplace-Transformation / Lineares Optimieren / Numerik gewöhnlicher DGLn (Runge-Kutta-, Galerkin-, Kollokationsverfahren,...) / partielle DGLn

ISBN 3-923923-05-8 244 Seiten ca. 18 DM

Günter Mühlbach

Repetitorium der Ingenieurmathematik Teil 3

Wahrscheinlichkeitsrechnung und Statistik :
Zufallsgrößen / Verteilungen / Korrelationen und Regressionen / Parameterschätzungen / Konfidenzintervalle / Qualitätskontrollen / Tests

ISBN 3-923923-30-9 174 Seiten ca. 18 DM

Franco Binomi

Vorbereitung zum Vordiplom — Mathematik für Ingenieure I,II

Für immer wieder auftretende Aufgabentypen in Vordiplomklausuren werden Lösungsrezepte angegeben. Zahlreiche typische Aufgaben werden ausführlich durchgerechnet.

ISBN 3-923923-11-2 78 Seiten ca. 8 DM

Gerhard Merziger / Thomas Wirth

BASIC - Programme zur höheren Mathematik

Hier sind über 60 Programme zu den grundlegenden Rechenverfahren — wie sie im **Repetitorium** behandelt werden — zusammengestellt, ausführlich kommentiert und mit Beispielen versehen.

ISBN 3-923923-15-5 192 Seiten ca. 17 DM

Disketten (Schneider 3" zu 24,80 DM / MSDOS 5,25" zu 17,80 DM) können beim Verlag bezogen werden.

Detlef Wille

Einführung in Basic

Dieses Buch ist für Programmieranfänger gedacht, die in Studium oder Beruf mathematische Aufgaben in BASIC-Programme umsetzen möchten.

ISBN 3-923923-20-1 128 Seiten ca. 13 DM

V o r w o r t

Zu den mathematischen Grundlagen, die Studenten von Natur- und Ingenieurwissenschaften bereits in den ersten Semestern kennenlernen, gehören eine ganze Reihe praktischer Rechenverfahren, angefangen bei der Nullstellenberechnung über die numerische Integration und Vektorrechnung bis zu den Differentialgleichungen.
Dieses Buch stellt über 60 BASIC-Programme zur höheren Mathematik - wie sie zum Beispiel in dem Arbeitsbuch.

REPETITORIUM DER INGENIEUR-MATHEMATIK

behandelt wird - zur Verfügung und will damit mehreren Zielen dienen:

(1) Durch ausführliche Anleitungen und reichliche Beispiele lernt der Benutzer, wie ein mathematisches Problem in einen vom Computer abzuarbeitenden Algorithmus umgesetzt wird.

(2) Der Benutzer erfährt, daß es mit einer exakten Problemanalyse und Programmierung zuweilen nicht getan ist: Die Computer-interne Zahlenverarbeitung (Rundungsfehler) zwingt zu steter Wachsamkeit und angemessenem Mißtrauen gegenüber den Ergebnissen.

(3) Der Benutzer kann rechenaufwendige praktische Aufgaben (z.B. E-Technik: lineares Gleichungssystem mit komplexen Koeffizienten) mit Hilfe des Computers bequem lösen.

Damit die Programme auf möglichst vielen Rechnern ohne Abänderungen lauffähig sind, haben wir nur die grundlegenden BASIC-Befehle verwendet. Sicher ließe sich manches kürzer und eleganter gestalten, wenn komfortablere Befehle des jeweils vorliegenden BASIC-Dialekts benutzt würden.

Die in dem Inhaltsverzeichnis ohne "Programmname auf Diskette" aufgeführten Kurzprogramme sind zum Abtippen für Taschencomputer, wie z.B. den SHARP 1401, **gedacht** und an dessen sprachliche Besonderheiten angepaßt, z.B.:

```
          50 IF x=0 THEN LET y=1
statt     50 IF x=0 THEN y=1

oder      60 PRINT "A(";K;")"
statt     60 PRINT "A("K")"
```

Kommentarzeilen, wie z.B.

```
         120 REM ***** GGT *****
         760 'RUNDEN
```

dienen nicht als Sprungadressen und können beim Abtippen ausgelassen werden.

Stellen Sie fest, wie Ihr Rechner auf '?PI' reagiert! Ist das Ergebnis 0, müssen Sie in den Programmen, die die Variable PI benutzen, PI definieren: z.B durch die Zeile

```
          5 PI=4*ATN(1)
```

In den betreffenden Programmen ist dies vermerkt durch die Fußnote *) gegebenenfalls PI=4*ATN(1) definieren!

Trotz aller Sorgfalt können sich Fehler eingeschlichen haben, so daß wir natürlich keine Garantie übernehmen können.
Für diesbezügliche Hinweise und Verbesserungsvorschläge sind wir dankbar.

Die Programmdiskette

BASIC - Programme
zur höheren Mathematik

ist beim Verlag in zwei Versionen erhältlich:

a) 3" Diskette für Schneider CPC (24,80 DM)
b) 5,25" Diskette für MS-DOS Rechner (17,80 DM)

Versand gegen Scheck oder Nachnahme

Inhaltsverzeichnis

			Programmname auf Diskette
(1)	**Näherungsweise Nullstellenbestimmung**	9	
	1 Newton	10	
	2 Newton	12	
	3 Newton (ohne Eingabe der Ableitung)	14	
	4 Nullstellen (mit Vorzeichenwechsel)	15	NULLST-1
	5 Nullstellen (mittels Ableitung)	18	NULLST-2
	6 Nullstellen	22	NULLST-3
(2)	**Polynome**	27	
	1 Rechnen mit Polynomen		
	Add, Mult, Div mit Rest, ggT, Kürzen	27	POLYNOME
	2 Hornerschema (reell)	33	HORNER-R
	3 Hornerschema (komplex)	33	HORNER-K
	4 Interpolation	35	
	Newtonsches Interpolationsverfahren	35	INTPOL
	Interpolationswerte	36	INTPOL-W
	5 Polynomzerlegung		
	Nullstellen eines Polynoms bis 5. Grad.		
	und Angabe der unzerlegbaren		
	quadratischen Faktoren	38	POLYZER
	6 Ausgleichskurven	42	
	Regressionsgerade	42	AUSGLGER
	Ausgleichspolynom	43	AUSGLPOL
(3)	**elementare Funktionen**	45	
	1 Definition durch Unterprogramme	45	ELEFUNK1
	2 Definition durch DEF FN	48	ELEFUNK2
(4)	**Schwingungen gleicher Frequenz**	49	
	1 Zerlegung und Überlagerung von		
	Schwingungen (SHARP)	50	
	2 Zerlegung und Überl. von Schwingungen	54	SCHWING
	3 Überlagerung von Schwing. (Bildschirm)	58	UEBERLAG

(5)	**Integration**	62	
	1 Integration (Gauß, Kurzprogramm)	62	
	2 Integration (Gauß, Kurzprogramm)	64	
	3 Integration (Gauß mit Definition elementarer Funktionen)	65	INTEGRAL
	4 Gebietsintegral (Doppelintegral)	67	INT-XY
(6)	**Partialbruchzerlegung**	70	PBZ
(7)	**lineare DGLn mit konstanten Koeffizienten**	84	
	1 allgemeine Lösung der hom. DGL, n≤5	84	DGLHOM
	2 eine spezielle Lös. der inh. DGL, n bel.	87	DGLINHOM
	3 allgemeine Lösung der inh. DGL, n≤5	91	DGL
(8)	**Runge-Kutta Verfahren zur Lösung von DGLn**	97	RUNGE
(9)	**lineare Gleichungssysteme (LGS)**	104	
	1 Gaußsches Eliminationsverfahren	104	LGS
	2 Kurzprogramm	107	
	3 Kurzprogramm bei eindeutiger Lösbarkeit	108	
	4 Pivot-Verfahren	109	LGSPIVOT
	5 LGS mit komplexen Koeffizienten	111	LGSKOMPL
(10)	**Matrizengleichung A•X = B**	114	MATR-GL
(11)	**Produkte und Potenzen von Matrizen**	117	
	1 Produkte	117	MATR-PRO
	2 Potenzen	118	MATR-POT
(12)	**Invertierung von Matrizen**	120	
	1 Invertierung	120	MATR-INV
	2 Kurzprogramm	122	
(13)	**Determinanten**	123	
	1 Berechnung der Determinante	123	DETERMIN
	2 Kurzprogramm	124	

(14)	**Eigenwerte und Eigenvektoren**	125	
	1 für 2x2-Matrizen	125	EIGWERT2
	2 für 3x3-Matrizen	128	EIGWERT3
	3 Bestimmung des charakterist. Polynoms	132	CHARPOL
	4 Eigenvektoren zu bekannten Eigenwerten	135	EIGVEKT
	5 Iterationsverfahren von Mises	137	MISES
(15)	**Vektorrechnung**	141	Vektor
(16)	**Rechnen mit komplexen Zahlen**	147	KOMPLEX
(17)	**quadratische Gleichung (komplex)**	152	QUADRGL
(18)	**Kegelschnitte**	153	
	1 Kegelschnitte	155	KEGEL
	2 Kurzprogramm	160	
(19)	**Bruchrechnung**	162	BRUCH
(20)	**ggT zweier ganzer Zahlen**	165	GGT
(21)	**Dezimalbruchentwicklung**	166	
	1 Division (Ergebnis mit beliebig vielen Nachkommastellen)	166	DEZBRU1
	2 Dezimalbruchentwicklung Vorperiode, Periode, Periodenlänge	166	DEZBRU2
(22)	**g - adische Zahlen**	168	
	1 Darstellung einer g-adischen Zahl als h-adische Zahl	168	G-ADISCH
	2 Kurzprogramm	171	
(23)	**Primzahlen**	172	
	1 Primzahltest	172	PRIMTEST
	2 Primfaktorzerlegung	172	PRIMZER
	3 Primzahlsieb (Eratosthenes)	173	PRIMSIEB

(24)	**Rechnen mit großen natürlichen Zahlen**	174	
	1 Multiplikation	174	NAT-MULT
	2 Division mit Rest	175	NAT-DIV
	3 Potenzen	176	NAT-POT
	4 Fakultäten	177	NAT-FAK
(25)	**komplexe Funktionen**	179	KOM-FUNK
(26)	**Gamma - Funktion**	184	GAMMA
(27)	**Funktionen in Wahrscheinlichkeitsrechnung und Statistik**	185	STATIST
	Index	190	

Die Programmdiskette

BASIC - Programme
zur höheren Mathematik

ist beim Verlag in zwei Versionen erhältlich:

a) 3" Diskette für Schneider CPC (24,80 DM)
b) 5,25" Diskette für MS-DOS Rechner (17,80 DM)

Versand gegen Scheck oder Nachnahme

Näherungsweise Nullstellenbestimmung

Das Newtonsche Näherungsverfahren ist wohl das bekannteste Verfahren, um Nullstellen näherungsweise zu berechnen. Da Tangenten benutzt werden, muß die Funktion differenzierbar sein (**REP** 136).

Hat eine stetige Funktion bei a einen positiven und bei b einen negativen Funktionswert, so liegt zwischen a und b eine Nullstelle, bei der die Funktion ihr Vorzeichen ändert. Diese ´Zwischenwerteigenschaft´ kann man benutzen, um durch fortgesetztes Halbieren des Intervalls [a,b] Nullstellen, bei denen die Funktion ihr Vorzeichen ändert, näherungsweise zu bestimmen (Programm **4**).

Um Nullstellen ohne Vorzeichenwechsel (Berührstellen) zu berechnen, kann man die Ableitung heranziehen (Programm **5**).

Will man ohne Ableitung auskommen, so kann man diese durch Funktionswertdifferenzen ersetzen (Programm **6**):

1 Newton (8 Programmzeilen)
Kurzprogramm des Newtonschen Näherungsverfahrens

2 Newton (11 bis 13 Programmzeilen)
Kurzprogramm, Iterationsschritte werden gezählt.

3 Newton (14 Programmzeilen)
Kurzprogramm, die Ableitung wird nicht programmiert.

4 Nullstellen (mit Vorzeichenwechsel) (17 Programmzeilen)
Kurzprogramm, berechnet Nullstellen mit Vorzeichenwechsel, Fehlerabschätzung

5 Nullstellen (mittels Ableitung) (55 Programmzeilen)
Berechnet alle Nullstellen, auch mehrfache ohne Vorzeichenwechsel. Die Ableitung muß programmiert werden.

6 Nullstellen (74 Programmzeilen)
Berechnet alle Nullstellen. Die Ableitung wird nicht benötigt. Die Funktion braucht nicht differenzierbar zu sein.

1 Newton

Folgendes kleine Programm berechnet nach dem Newtonschen Näherungsverfahren (**REP** 136, **B** 782) fortlaufend - nach Betätigen der ENTER-Taste - Näherungen.

Die Funktion f(x) und ihre Ableitung f'(x) müssen in Zeile 60 bzw. Zeile 70 programmiert werden.

Einzugeben nach dem Programmstart:

x_0 der Startwert

```
10 INPUT "x0=";x                    Eingabe Startwert
20 GOSUB 60                         holt f(x) und f´(x)
30 x=x-f/g                          die nächste Näherung
40 PRINT x                          Ausgabe der Näherung
50 GOTO 20          noch einmal, nach Drücken der ENTER-Taste
60 f=                               Berechnung des Funktionswertes
70 g=                           Berechnung des Wertes der Ableitung
80 RETURN             zurück zu Zeile 20, weiter mit Zeile 30
```

Beispiel 1: Nullstellen von $f(x) = x^2-2$

Die exakten Nullstellen sind $\pm\sqrt{2} = \pm\ 1{,}414213562$

Wir programmieren die Funktion f(x) in Zeile 60:

```
60 f=x*x-2
```

und die Ableitung f´(x) in Zeile 70:

```
70 g=2*x
```

Nach dem Programmstart wird eingegeben:

```
x0=1        ENTER       1.5
            ENTER       1.416666667
            ENTER       1.414215686
            ENTER       1.414213562
```

Ein genaueres Ergebnis können wir nicht erwarten. So schnell muß es natürlich nicht immer gehen:

```
x0=-0.3     ENTER      -3.483333333
            ENTER      -2.028748006
            ENTER      -1.507288844
            ENTER      -1.417087268
```

Beispiel 2: Nullstelle von $f(x) = \cos x - \sqrt{x}$

```
60 f=cos(x)-√x
70 g=-sin(x)-1/2/√x
```

```
x0=1        ENTER       6.573181983 E-01
            ENTER       6.417461053 E-01
```

Haben Sie vergessen, auf RADIAN umzuschalten, erhalten Sie nach ENTER 9.997056641 E-01.

Beispiel 3: Man bestimme die im Intervall $(\pi, 2\pi)$ gelegene Lösung von $\tan x - x = 0$

```
60 f=tan(x)-x
70 g=1/cos(x)/cos(x)-1
```

```
x0=3    ENTER       1076,188203     wieder RAD vergessen!
x0=3    ENTER        157,6565578    zu weit
x0=4    ENTER          6,120158487  schon besser,aber
        ENTER        238,4042778
x0=5    ENTER          5,733339321
        ENTER         22,62787283   so nicht!
```

Erst einmal überlegen: Die gesuchte Lösung liegt zwischen π und $1{,}5\,\pi$ (Skizze!!!). Versuchen wir also:

```
x0=4.5  ENTER          4.493613903
        ENTER          4.493409655
        ENTER          4.493409458
        ENTER          4.493409458   das wär´s.
```

Beispiel 4: Versuchen wir, eine mehrfache Nullstelle zu finden: $f(x) = x^3$

```
60 f=x*x*x
70 g=3*x*x
```

```
x0=1    ENTER          6.666666667 E-01
        ENTER          4.444444444 E-01
        ENTER          2.962962963 E-01    etwas langsam
```

Beispiel 5: Dieses Beispiel zeigt deutlich die Schwäche des Newton-Verfahrens:Bei mehrfachen Nullstellen ist es sehr langsam:

Nullstellen von $f(x) = x^5$

```
60 f=x↑5
70 g=5*x↑4
```

```
x0=0.1  ENTER          0.08
        ENTER          0.064
        ENTER          0.0512        und so weiter
```

2 Newton

Folgendes Programm führt das Newton-Verfahren durch, bis sich zwei Näherungen um weniger als 10^{-6} unterscheiden. Die Anzahl der durchgeführten Iterationsschritte wird gezählt und die Näherung ausgegeben. Nach Betätigen der ENTER-Taste wird die Anzahl der durchgeführten Iterationen angezeigt.

Die Funktion f(x) und ihre Ableitung f´(x) müssen in Zeile 60 bzw. Zeile 70 programmiert werden.

Durch die zusätzlichen Zeilen 15 und 45 werden bei Erreichen einer vorgegebenen Schrittzahl nochmals ein (besserer?) Startwert und eine (größere?) Schrittzahl verlangt.
Dadurch werden hoffnungslose Versuche frühzeitig abgebrochen, wie es Beispiel 7 zeigt.

```
10 INPUT "x0=";x:k=0
15 INPUT "Schrittzahl=";n
20 GOSUB 60
30 u=x-f/g:k=k+1
40 IF ABS(u-x)<0.000001 THEN 90
45 IF k>n-1 THEN 10
50 x=u:GOTO 20
60 f=
70 g=
80 RETURN
90 PRINT u
100 PRINT"k=";k
110 END
```

Beispiel 6: (siehe Beisp. 1) Nullstellen von $f(x) = x^2-2$

```
60 f=x*x-2
70 g=2*x
```

ohne Zeilen 15,45:

```
x0=1      ENTER          1.414213562
          ENTER          k=5
```

mit Zeilen 15,45:

```
x0=1                ENTER
Schrittzahl=6       ENTER    1.414213562
                    ENTER    k=5

x0=-2               ENTER
Schrittzahl=4       ENTER    Die Schrittzahl reicht nicht, um die
                             gewünschte Genauigkeit zu erreichen.
x0=-2               ENTER    Es werden neue Werte verlangt!
Schrittzahl=10      ENTER    -1.414213562
                    ENTER    k=5
```

Beispiel 7: Man bestimme die im Intervall $(\pi, 2\pi)$ gelegene Lösung von $\tan x - x = 0$

(s. Bsp.3)

```
60 f=tan(x)-x
70 g=1/cos(x)/cos(x)-1
```

ohne Zeilen 15,45:

```
x0=3      ENTER      nach ca. 20 sek "ERROR 2 IN 60"
```

mit Zeilen 15,45:

```
x0=3                ENTER
Schrittzahl=10      ENTER    nach 10 Schritten werden neue
                             Werte verlangt:
x0=4                ENTER
Schrittzahl=10      ENTER    reicht noch nicht!
x0=4.5              ENTER
Schrittzahl=5       ENTER    4.493409458
                    ENTER    k=3
```

Beispiel 8: Nullstelle von $f(x) = x^3$

(s. Bsp. 4)

```
60 f=x*x*x
70 g=3*x*x
```

ohne Zeilen 15,45:

```
x0=1                ENTER        1.545213349 E-01
                    ENTER        k=33
```

mit Zeilen 15,45:

```
x0=1
Schrittzahl=10      ENTER                reicht nicht
x0=1                ENTER
Schrittzahl=20      ENTER                immer noch nicht
x0=0.001            ENTER
Schrittzahl=20      ENTER        1,522438839 E-01
                    ENTER        k=16
```

Beispiel 9: Nullstelle von $f(x) = x^5$

(s. Bsp. 5)

```
60 f=x↑5
70 g=5*x↑4
```

ohne Zeilen 15,45:

```
x0=0.1              ENTER        3.741444191 E-06
                    ENTER        k=56
```

mit Zeilen 15,45:

```
x0=0.1              ENTER
Schrittzahl=20      ENTER                reicht nicht
x0=0.0001           ENTER
Schrittzahl=20      ENTER        3.518437209 E-06
                    ENTER        k=15
```

Obwohl der Startwert schon eine gute Näherung ist, werden noch 15 Iterationsschritte benötigt!

3 Newton

Bei den vorigen Programmen mußte die Ableitung eingegeben werden. Die manchmal mühsame Berechnung der Ableitung kann vermieden werden, wenn die Ableitung durch den Differenzenquotienten

$$f'(x) \approx \frac{f(x+10^{-8}) - f(x)}{10^{-8}}$$

ersetzt wird (siehe z.B. **DE** 391). Folgendes Programm führt das Newton-Verfahren wie das Programm **2 Newton** durch. Lediglich die Funktion f(x) muß in Zeile 100 progammiert werden:

```
10 INPUT "x0=";x:k=0
20 INPUT "Schrittzahl=";n
30 GOSUB 100
40 e=f:x=x+10↑(-8)
50 GOSUB 100
60 u=x-e*10↑(-8)/(f-e):k=k+1
70 IF ABS(u-x)<10↑(-6) THEN 120
80 IF k>n-1 THEN 10
90 x=u:GOTO 30
100 f=
110 RETURN
120 PRINT u
130 PRINT"k=";k
140 END
```

Beispiel 10: (siehe Beisp. 1)

Nullstellen von $f(x) = x^2-2$

```
100 f=x*x-2
```

```
    x0= 1                ENTER
    Schrittzahl= 10      ENTER        1.41421357
                         ENTER        k=6
```

Vorteil: Die manchmal mühsame Berechnung von f'(x) entfällt.
Nachteil: Noch langsamer als **2 Newton**.

4 Nullstellen (mit Vorzeichenwechsel) *)

Hat die Funktion f bei a und b verschiedene Vorzeichen, d.h. ist f(a)*f(b)<0, so liegt zwischen a und b eine Vorzeichenwechselstelle' von f. Diese ist bei stetigen Funktionen eine Nullstelle (Zwischenwerteigenschaft stetiger Funktionen). Bei unstetigen Funktionen kann es z.B. eine Sprungstelle oder eine Polstelle sein.
Hat die (stetige) Funktion f zwischen a und b eine Nullstelle mit Vorzeichenwechsel, so kann man das Intervall [a,b] fortlaufend halbieren, die Vorzeichen der Funktionswerte an den Intervallrändern untersuchen und so ein Näherungsverfahren für eine Nullstelle von f zwischen a und b erhalten.

Bei gegebener Schrittzahl n (Anzahl der Halbierungen) ist der Fehler, d.h. der Abstand von Näherung und exakter Nullstelle, kleiner als $(b-a)/2^n$.
Ist die Intervallänge z.B. kleiner als 8, so reicht n=23, um den Fehler kleiner als 0,000001 zu machen!

Die Funktion f(x) ist in Zeile 140 zu programmieren.
Nach dem Programmstart sind einzugeben:

Die untere Grenze a=
Die obere Grenze b=
Die Schrittzahl n=

```
10 INPUT "a=";a
20 INPUT "b=";b
30 INPUT "Schrittzahl n=";n
40 x=a:GOSUB 140:u=f
50 x=b:GOSUB 140:v=f
60 FOR k=1 TO n
70 x=(a+b)/2:GOSUB 140:w=f
80 IF u*w>0 THEN LET a=x:u=w:GOTO 100
90 b=x:v=w
100 NEXT k
110 PRINT x
120 END
130 REM ***** Eingabe von f(x) *****
140 f=
150 IF f=0 THEN 110
160 f=SGN(f)
170 RETURN
```

*) gegebenenfalls Pi=4*ATN(1) definieren

Beispiel 11: Nullstelle von $f(x) = x^2+x-2$

zwischen 0 und 5

Zwischen 0 und 5 liegt eine Nullstelle, da f(0)=-2 und f(5)=28 verschiedene Vorzeichen haben.
Wir wählen n=23 , um den Fehler kleiner als 0,000001 zu machen.

```
140 f=x*x+x-2     RUN
a=0               ENTER
b=5               ENTER
Schrittzahl n=23  ENTER        0.999999642
```

Es ist $x^2+x-2 = (x+2)(x-1)$ und die gesuchte Nullstelle also 1.
Obiges Ergebnis ist nicht voll zufriedenstellend.
Wir fügen einige Programmschritte ein, um das Ergebnis zu runden:

```
105 GOSUB 190
180 REM ***** Runden *****
190 x=x*1000:y=INT(x)
200 IF x-y<0.001 THEN LET x=y:GOTO 220
210 IF y+1-x<0.001 THEN LET x=y+1
220 x=x/1000
230 RETURN
```

Mit dieser Ergänzung des obigen Programms erhalten wir als Näherung den exakten Wert 1.

Beispiel 12: Nullstellen von $f(x) = \sin(x^2-4)$

zwischen 0 und 3

Es ist $\sin(0^2-4)=\sin(-4)<0$ und $\sin(3^2-4)=\sin(5)<0$, also liegt (mindestens) eine Nullstelle von f zwischen 0 und 3:

```
a=0
b=3
n=23              ergibt   0.926502586
```

Haben Sie eine andere Nullstelle erwartet? Vielleicht 2?
Suchen wir weiter:

Es ist $\sin(1^2-4)<0$ und $\sin(2{,}5^2-4)>0$, also liegt eine Nullstelle von f zwischen 1 und 2,5:

a=1
b=2.5
n=23 ergibt 2

Da $\sin(2{,}5^2-4)>0$ und $\sin(3^2-4)<0$ ist, muß zwischen 2,5 und 3 noch eine Nullstelle liegen:

a=2.5
b=3
n=23 ergibt 2.67237586

Dies sind übrigens alle Nullstellen von f zwischen 0 und 3, denn $\sin(x^2-4)=0 \iff x^2-4=k\pi \iff x^2=4+k\pi \iff x=\sqrt{4+k\pi}$ und $\sqrt{4+k\pi}$ ergibt genau für k=-1,0,1 Werte zwischen 0 und 3.

Beispiel 13: (siehe **Polynomzerlegung** Beispiel 1)

Nullstellen von $f(x) = x^4+6x^3+16x^2+26x+15$
zwischen -5 und 0

Es ist $f(-5)>0$ und $f(-2)<0$ und $f(0)>0$:

a=-5
b=-2
n=23 ergibt -3

a=-2
b=0
n=23 ergibt -1

Wir haben also zwei Nullstellen gefunden, könnten jetzt die beiden Linearfaktoren (x+3) und (x+1) abspalten (z.B. **Horner**) und eine quadratische Gleichung lösen, um die eventuell vorhandenen weiteren reellen Nullstellen zu finden.

Dieses Programm ist kurz und im Vergleich zu **Newton** wesentlich schneller.
Es lassen sich jedoch nur Nullstellen mit Vorzeichenwechsel finden. Nullstellen ohne Vorzeichenwechsel, also Nullstellen, bei denen die x-Achse berührt wird, ohne daß sich das Vorzeichen der Funktionswerte bei der Nullstelle ändert, sind schwieriger zu finden.

5 Nullstellen (mittels Ableitung) *)

Dieses Programm berechnet alle Nullstellen der Funktion f zwischen a und b. Mehrfache Nullstellen im Innern oder auf dem Rand des Intervalls (a,b) mit und ohne Vorzeichenwechsel werden gefunden.

Das Programm benutzt das Halbierungsverfahren auch für die Ableitung f´(x).

Die Funktion f(x) muß in Zeile 100 und die Ableitung f´(x) in Zeile 120 (als g) programmiert werden.

```
10 CLS
20 PRINT"Naeherungsverfahren zur Berechnung"
30 PRINT"der Nullstellen von f zwischen a und b"
40 PRINT:PRINT
50 INPUT "f(x),f´(x) eingegeben (j/n)";e$
60 PRINT:IF ASC(e$)>106 THEN LIST 100-120
70 PRINT
80 REM *** Eingabe von f,f´ und a,b ***
90 GOTO 140
100 f=
110 y=SGN(f):RETURN
120 g=
130 y=SGN(g):RETURN
140 INPUT "a,b";a,b:PRINT
150 m=0
160 REM*******************************
170 REM *** Vorzeichenwechsel suchen ***
180 h=1
190 c=10↑h*(b-a):d=b:b=a
200 FOR l=0 TO c
210 a=b:b=a+10↑(-h)
220 IF b>d THEN LET b=d
230 FOR j=1 TO 2
240 x=a:ON j GOSUB 100,120:u=y
250 IF u=0 THEN GOSUB 400:GOTO 280
```

*) gegebenenfalls Pi=4*ATN(1) definieren

```
260 x=b:ON j GOSUB 100,120:v=y
270 IF u*v<0 THEN LET p=a:q=b:GOSUB 330:GOSUB 400
280 NEXT j:NEXT l
290 IF m=0 THEN PRINT"keine Nullstellen zwischen a und b"
300 PRINT:END
310 REM ******************************
320 REM ***** Halbierungen *****
330  FOR k=1 TO 20:x=(p+q)/2
340 ON j  GOSUB 100,120:w=y
350 IF u*w>0 THEN LET p=x:u=w:GOTO 370
360 q=x:v=w
370 NEXT k
380 RETURN
390 REM ***** f(x)=0 ? *****
400           GOSUB 100
410 IF ABS(f)<10↑(-3) THEN GOSUB 450
420 j=2
430 RETURN
440 REM ***** Ausgabe *****
450 m=m+1
460 GOSUB 500
470 PRINT"x(";m;")=";x
480 RETURN
490 REM **** Runden *****
500 x=x*1000
510 y=INT(x)
520 IF x-y<0.001 THEN LET x=y:GOTO 540
530 IF y+1-x<0.001 THEN LET x=y+1
540 x=x/1000
550 RETURN
```

Beispiel 14: (siehe **REP**,49 Bsp 13,18)

Nullstellen von $f(x) = x^4-5x^3+5x^2+5x-6$ zwischen -5 und 5

```
100 f=x↑4-5*x↑3+5*x*x+5*x-6
120 g= 4*x↑3-15*x*x+10*x+5
```

a,b=-5,5 ergibt:

x(1) = -1
x(2) = 1
x(3) = 2
x(4) = 3

Beispiel 15: Nullstellen von $f(x) = x^5(x-1)^6$ zwischen -1 und 2

Wir berechnen mit der Produktregel (**REP**,129) die Ableitung:
$f'(x) = 5x^4(x-1)^6 + x^5 6(x-1)^5 = x^4(x-1)^5(11x-5)$,
setzen f in Zeile 100 und f' für g in Zeile 120 ein und starten das Programm:

```
a,b=-1,2        ergibt
x(1) = 0
x(2) = 1
```

Mehrfache Nullstellen mit und ohne Vorzeichenwechsel werden gefunden

```
a,b=0,1         ergibt
x(1) = 0
x(2) = 1
```

Nullstellen auf dem Rand werden gefunden

Dennoch: Überraschungen kann es immer geben:

Beispiel 16: (siehe Bsp 3,7)

Nullstelle von $f(x) = \tan(x) - x$ zwischen 3 und 6

Wir setzen f(x) und $f'(x) = 1/\cos^2(x)-1$ in den Zeilen 100 und 120 ein:

```
a,b=3,6         ergibt
x(1) = 4.49340944
Division by zero
```

Nanu? $f(x)=\tan x - x$ hat bei $\frac{3}{2}\pi = 4{,}71238898$ eine Polstelle mit Vorzeichenwechsel. Diese Stelle wird gefunden, aber nicht ausgegeben.
Beim Abfragen "f(x)=0" ab Programmzeile 390 tritt natürlich der Fehler "Division by zero" auf.
Solche Überraschungen vermeiden wir, wenn wir uns auf solche Intervalle [a,b] beschränken, auf denen f differenzierbar ist:

```
a,b = 3,4.7       ergibt
x(1) = 4.49340944
```

Beispiel 17: (siehe Bsp 12)

Nullstellen von $f(x) = \sin(x^2-4)$
zwischen -4 und 4

Wir setzen f(x) und $f'(x) = 2x\cos(x^2-4)$ in den Zeilen 100 und 120 ein und starten das Programm:

a,b=-4,4 ergibt

x(1) = -3.66398383
x(2) = -3.20674067
x(3) = -2.67237597
x(4) = -2
x(5) = -0.926502903
x(6) = 0.9265027
x(7) = 2
x(8) = 2.67237577
x(9) = 3.20674066
x(10)= 3.66398381

Damit haben wir alle Nullstellen von f(x) zwischen -4 und 4 gefunden.

Das Programm ist relativ kurz, zumal die ersten 7 Zeilen und die REM-Zeilen bei Benutzung eines Taschenrechners weggelassen werden können.
Das Programm eignet sich besonders zum Auffinden von Nullstellen von Funktionen, deren Ableitung sich leicht berechnen läßt.
Will man auf das manchmal mühsame Berechnen der Ableitung verzichten, kann man das folgende Programm benutzen: Statt Vorzeichenwechsel der Ableitung zu suchen, sucht es solche der Funktionswertdifferenzen. Das hat den zusätzlichen Vorteil, daß die Funktion nicht differenzierbar zu sein braucht. So können auch Funktionen, in denen die Betragsfunktion vorkommt, untersucht werden.

6 Nullstellen *)

Dieses Programm berechnet Näherungen für alle Nullstellen einer Funktion f(x) zwischen a und b.

Die Funktion f(x) ist in Zeile 290 zu programmieren.

```
10 CLS
20 REM ***** Eingabe *****
30 PRINT"  Naeherungsverfahren zur Berechnung":PRINT
40 PRINT"der Nullstellen von f zwischen a und b":PRINT:PRINT
50 INPUT "f(x) in Zeile 290 eingegeben (j/n)";e$
60 PRINT:IF ASC(e$)>106 THEN LIST 290
70 e=1
80 INPUT "a,b=";a,b
90 DIM x(30)
100 REM *** Vorzeichenwechsel suchen ***
110 x=a:z=a:m=0:h=10↑(-7):s=10↑(-e)
120 GOSUB 290:u=g
130 IF u=0 THEN GOSUB 520
140 x=x+h:GOSUB 320:v=g
150 FOR j=1 TO 2
160 p=z:z=z+s
170 IF z>b THEN LET z=b
180 x=z: ON j GOSUB 290,320:y=g
190 IF y=0 THEN ON j GOSUB 520,480:GOTO 210
200 IF u*y<0 THEN GOSUB 370
210 IF z<b THEN LET u=y:GOTO 160
220 z=a:u=v
230 NEXT j
240 PRINT
250 IF m=0 THEN PRINT"keine Nullstellen zwischen a und b":GOTO 270
260 GOSUB 560:PRINT
270 END
280 REM ***** f und Diff-Quotient *****
290 f=
300 g=SGN(f)
310 RETURN
320 GOSUB 290:d=f
```

*) gegebenenfalls Pi=4*ATN(1) definieren

```
330 x=x-h:GOSUB 290:x=x+h
340 g=SGN(d-f)
350 RETURN
360 REM ***** Halbierungen *****
370 q=x
380 FOR k=1 TO 20:x=(p+q)/2
390 ON j  GOSUB 290,320
400 IF g=0 THEN LET k=20:GOTO 450
410 IF u*g>0 THEN LET p=x:u=g:GOTO 430
420 q=x
430 NEXT k
440 u=g
450 GOSUB 480
460 RETURN
470 REM ***** f(x)=0 ? *****
480 GOSUB 660:GOSUB 290
490 IF ABS(f)<10↑(-3) THEN GOSUB 520
500 RETURN
510 REM ***** Nullstellen sammeln *****
520 m=m+1
530 x(m)=x
540 RETURN
550 REM ***** Ausgabe *****
560 n=1
570 FOR k=1 TO m
580 FOR l=k+1 TO m
590 IF ABS(x(k)-x(l))<10↑(-4) THEN LET l=m:GOTO 630
600 NEXT l
610 x=x(k):GOSUB 660:x(k)=x
620 PRINT"x(";n;") = ";x(k):n=n+1
630 NEXT k
640 RETURN
650 REM **** Runden *****
660 x=x*1000:i=INT(x)
670 IF x-i<0.001 THEN LET x=i:GOTO 690
680 IF i+1-x<0.001 THEN LET x=i+1
690 x=x/1000
700 RETURN
```

Beispiel 18: (siehe **REP**, 49 Bsp. 13)

Nullstellen von $f(x) = x^4-5x^3+5x^2+5x-6$ zwischen -5 und 5

Wir programmieren die Funktion f(x) in Zeile 290:

```
290 f=x*x*x*x-5*x*x*x+5*x*x+5*x-6
```

und starten das Programm:

a,b = -5,5 ergibt

x(1) = -1
x(2) = 1
x(3) = 2
x(4) = 3

Beispiel 19: Nullstellen von $f(x) = x^5-3x^4-10x^3+30x^2+9x-27$ zwischen -5 und 5

a,b = -5,5 ergibt

x(1) = -3
x(2) = -1
x(3) = 1
x(4) = 2.99996233
x(5) = 3.00003996

die exakten Nullstellen sind -3,-1,1,3,3.

Bei folgendem Beispiel liegen die Nullstellen auf dem Rand des Definitionsbereiches von f. Außerdem ist die Steigung am Rand unendlich groß.
Versuchen Sie dieses Beispiel mal mit **5 Nullstellen**!

Beispiel 20: Nullstellen von $f(x) = \sqrt{1-x^2}$ zwischen -1 und 1

a,b = -1,1 ergibt

x(1) = -1
x(2) = 1

Im folgenden Beispiel wird eine stetige aber nicht differenzierbare Funktion auf Nullstellen untersucht:

Beispiel 21: alle Nullstellen von $f(x) = \sqrt{|1-x^4| + x^2-1}$

Wir programmieren:

```
290 f=sqr(abs(1-x↑4)+x*x-1)
```

Damit $f(x)=0$ ist, muß, da $|1-x^4| \geq 0$ ist, $x^2-1 \leq 0$ sein, also x zwischen -1 und 1 liegen:

a,b = -1,1 ergibt

x(1) = -1
x(2) = 1
x(3) = 0

Beispiel 22: alle Nullstellen von $f(x) = \sqrt{|(x-1)(x^2-2)|}$ zwischen -2 und 2

Wir programmieren

```
290 f=sqr(abs((x-1)*(x*x-2)))
```

a,b=-2,2 ergibt

x(1) = -1.41421356
x(2) = 1
x(3) = 1.41421357

Auch bei diesem Beispiel sieht man die Nullstellen sofort. In der Praxis würde man statt f(x) natürlich $f^2(x) = |(x-1)(x^2-2)|$ auf Nullstellen untersuchen, zumal in der Nähe einer Nullstelle die Funktionswerte durch das Wurzelziehen relativ groß werden. Zum Beispiel hat $\sqrt[3]{x}$ sicher bei 0 eine Nullstelle und x=0,000001 ist eine gute Näherung; aber $\sqrt[3]{x} = \sqrt[3]{0,000001} = 0,01$ und folglich würde x nicht als Nullstelle erkannt.

Das dem Programm zugrunde liegende "Vorzeichenwechsel-Such-Verfahren" geht mit der Schrittweite 0,1 vor. Liegen Nullstellen dichter beisammen, werden möglicherweise welche übersehen. Durch Änderung des Parameters e=1 in Zeile 70 in e=2,... kann die Schrittweite in 0,01 ... geändert werden.
Das Vergrößern des Parameters e erhöht natürlich die Rechenzeit!

Die Funktion $f(x)=\sin\frac{\pi}{x}$ ist für alle $x\neq 0$ erklärt und hat die positiven Nullstellen 1, $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$,... . Es gibt also beliebig dicht beieinander liegende Nullstellen. Damit kann man natürlich jedes Nullstellen-Näherungsprogramm in Verlegenheit bringen:

Beispiel 23: alle Nullstellen von $f(x) = \sin\frac{\pi}{x}$ für $x > 0{,}145$

```
290 f=sin(PI/x)                    *)
a,b = 0.145,1      ergibt
x(1) = 0.5
x(2) = 1
```

Es ist: $\frac{1}{7} = 0{,}142 < 0{,}145 < 0{,}16666 = \frac{1}{6}$

Von den sechs -nämlich $\frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1$ - zwischen 0,145 und 1 gelegenen Nullstellen haben wir nur zwei erhalten. Das liegt an der Schrittweite 0,1 beim Suchen von Vorzeichenwechseln.
Bei den ersten zwei Schritten (von 0,145 nach 0,245 und von 0,245 nach 0,345) werden jeweils zwei (einfache) Nullstellen überschritten: Kein Vorzeichenwechsel!

Durch Änderung von Zeile 70 e=2 (statt e=1) verkleinern wir die Schrittweite auf 0,01:

```
70 e=2
a,b=0.145,1   ergibt
x(1) = 0.166666665
x(2) = 0.2
x(3) = 0.25
x(4) = 0.333333331
x(5) = 0.5
x(6) = 1        , also alle Nullstellen!
```

Die Änderung e=2 erhöht natürlich die Rechenzeit (beim Schneider z.B Faktor 3).

---

*) gegebenenfalls PI = 4*ATN(1) definieren!

1 Rechnen mit Polynomen

Ebenso wie die Programme 'Bruchrechnung' und 'Komplexe Zahlen' ist dieses Programm daraufhin angelegt, Kettenrechnung zu ermöglichen. Problemlos ist ein Zusammenfassen von z.B.

$$\frac{(2x^2-3x+1)(3x^2-1)+(x-1)^3(2x-3)}{(2x-1)^2(x^2-1)+x^3-1}$$

durchzuführen.

Die eingegebenen Polynomkoeffizienten werden auf einem 'Stapel' abgelegt und eine nachfolgende Rechenoperation verknüpft die beiden zuletzt eingegebenen Koeffizientensätze miteinander.

Folgende Eingaben in den Rechner sind möglich :

(**RET** heißt hier : RETURN- oder ENTER-Taste drücken,
p heißt : p-Taste drücken)

1) Polynomkoeffizienten

Die Koeffizienten sind in absteigender Reihenfolge (der zugehörigen Exponenten) einzugeben und **die Eingabe ist durch Drücken der p-Taste abzuschließen.**

Beispiel : $p(x) = 5x^4 - 2x^3 + 7x - 12$
Eingabe : 5 **RET** -2 **RET** 0 **RET** 7 **RET** -12 **RET** p **RET**

2) Rechenoperationen + , - , * , / , ↑k (k natürliche Zahl)

Beispiel $p(x)=(x^2-3x+4)(x^3-3)$

Eingabe : 1 **RET** -3 **RET** 4 **RET** p **RET** 1 **RET** 0 **RET** 0 **RET** -3 **RET**
p **RET** * **RET**

Ausgabe : 1 -3 4 -3 9 -12
Ergebnis : $p(x) = x^5 - 3x^4 + 4x^3 - 3x^2 + 9x - 12$

Beispiel : $p(x)=(x^6+12x^4-5x^3-5x^2+5x-5)/(x^3+1)$

Eingabe : 1 **RET** 0 **RET** 12 **RET** -5 **RET** -5 **RET** 5 **RET** -5 **RET**
p RET 1 **RET** 0 **RET** 0 **RET** 1 **RET** p **RET** / **RET**

Ausgabe : Polynomanteil s(3)=1 s(2)=0 s(1)=12 s(0)=-6
Rest r(2)=-5 r(1)=-7 r(0)=1

Ergebnis : $p(x) = x^3 + 12x - 6 + \dfrac{-5x^2 - 7x + 1}{x^3 + 1}$

Beispiel : $p(x) = (2x-1)^2(3x+1)(4x-1)^3$.

Eingabe : 2 **RET** -1 **RET** p **RET** ↑2 **RET** 3 **RET** 1 **RET** p **RET** * **RET**
4 **RET** -1 **RET** p **RET** ↑3 **RET** * **RET**

Ergebnis : $p(x) = 768x^6-1088x^5+464x^4+4x^3-52x^2+13x-1$

3) Rechenanweisungen ggt , k

a) ggt : Es wird der GGT der beiden letzten zuvor eingegebenen Polynome bestimmt.

Beispiel : GGT($x^6+2x^4-3x^2-6$, $x^5+x^3+x^2-2x+2$)

Eingabe : 1 **RET** 0 **RET** 2 **RET** 0 **RET** -3 **RET** 0 **RET** -6 RET
p **RET** 1 **RET** 0 **RET** 1 **RET** 1 **RET** -2 **RET** 2 **RET**
p **RET** ggt **RET**

Ergebnis : $x^2 + 2$ ist GGT.

b) k : Die beiden letzten zuvor eingegebenen Polynome werden durch ihren GGT gekürzt.

Beispiel : $p(x) = \dfrac{6x^5 + 10x^4 + 13x^3 + 8x^2 + 2x + 3}{2x^4 + 10x^3 + 21x^2 + 22x + 15}$

Eingabe : 6 **RET** 10 **RET** 13 **RET** 8 **RET** 2 **RET** 3 **RET** p **RET**
2 **RET** 10 **RET** 21 **RET** 22 **RET** 15 **RET** p **RET** k **RET**

Ergebnis : $x^2 + x + 1.5$ ist GGT und

$$p(x) = \frac{6x^3 + 4x^2 + 2}{2x^2 + 8x + 10}$$

4) Weitere Anweisung : a

Nach Drücken der Taste 'a' und **RET** werden die Koeffizienten in den 4 Stapelregistern angezeigt.

```
10 CLS:PRINT"Rechnen mit Polynomen"
20 d=20:g=-1:DIM z(3,d),p(d),q(d),e(d),r(d),s(d):t=5
30 INPUT ;o$:IF o$="e" THEN END
40 IF o$="p" THEN GOSUB 160:PRINT:GOTO 30
50 IF o$="+" THEN 260
60 IF o$="-" THEN 260
70 IF o$="*" THEN 220
80 IF o$="/" THEN 320
90 IF o$="k" THEN 1090
100 IF LEFT$(o$,1)="↑" THEN 420
110 IF o$="a" THEN 530
120 IF o$="ggt" THEN 860
130 g=g+1:e(g)=VAL(o$):GOTO 30
140 '
150 ' Ablage in das 1.Register
160 FOR i=3 TO 1 STEP -1:g(i)=g(i-1):FOR p=0 TO d
170 z(i,p)=z(i-1,p):NEXT p:NEXT i
180 FOR i=0 TO g:z(0,i)=e(g-i):NEXT i
190 FOR i=g+1 TO g(0):z(0,i)=0:NEXT i:g(0)=g:g=-1:RETURN
200 '
210 ' *
220 FOR i=0 TO g(1):p(i)=z(1,i):NEXT i:n=g(1)
230 FOR i=0 TO g(0):q(i)=z(0,i):NEXT i:m=g(0):g=n+m
240 GOSUB 1030:GOTO 270
250 ' + -
260 g=g(0):IF g<g(1) THEN g=g(1)
270 FOR i=0 TO g:e(i)=r(i)
280 IF o$="+" THEN e(i)=z(0,i)+z(1,i)
290 IF o$="-" THEN e(i)=z(1,i)-z(0,i)
300 NEXT i:GOTO 580
310 ' /
320 GOSUB 820:GOSUB 720
330 PRINT:PRINT"Der Polynomanteil hat den Grad"l
340 FOR i=l TO 0 STEP-1
350 PRINT"s("i")="s(i):NEXT i
360 IF k=0 AND ABS(p(0))<10↑(-5) THEN PRINT"Rest 0":GOTO 30
```

```
370 PRINT"Der Rest hat den Grad"k
380 FOR i=k TO 0 STEP-1
390 PRINT"r("i")="p(i):NEXT i:GOTO 30
400 '
410 ' ↑k
420 kk=VAL(RIGHT$(o$,LEN(o$)-1)):h=1
430 m=g(0)
440 FOR i=0 TO m:q(i)=z(0,i):NEXT i
450 n=g(0)
460 FOR i=0 TO n:p(i)=z(0,i):NEXT i
470 GOSUB 1020:h=h+1
480 g(0)=n+m:FOR i=0 TO g(0):z(0,i)=r(i):NEXT i
490 IF h=kk THEN PRINT:GOTO 630
500 GOTO 450
510 '
520 ' Anzeige aller Register
530 PRINT:FOR k=0 TO 3
540 PRINT k+1"tes Reg.:";:FOR p=g(k) TO 0 STEP-1
550 PRINT z(k,p);:NEXT p:PRINT:NEXT k:GOTO 30
560 '
570 ' Herunterziehen von 3.und 4.Register
580 PRINT:FOR j=1 TO 2:g(j)=g(j+1):FOR i=0 TO d
590 z(j,i)=z(j+1,i):NEXT:NEXT:GOSUB 660
600 FOR i=0 TO g:z(0,i)=e(i):NEXT i
610 FOR i=g+1 TO g(0):z(0,i)=0:NEXT i:g(0)=g:g=-1
620 ' Anzeige 1.Register
630 FOR p=g(0) TO 0 STEP-1:PRINT z(0,p);:NEXT p:PRINT:GOTO 30
640 '
650 ' Gradtest
660 FOR p=g TO 0 STEP -1
670 IF e(p)=0 THEN NEXT:g=0:GOTO 690
680 g=p
690 RETURN
700 '
710 ' Division p(x)/q(x)=s(x) Rest r(x)
720 l=n-m:IF l<0 THEN PRINT"echt gebrochen":GOTO 30
730 FOR j=0 TO l:s(l-j)=p(n-j)/q(m)
```

```
740 FOR i=l-j TO n-j
750 p(i)=p(i)-s(l-j)*q(i-l+j):NEXT i:NEXT j
760 k=m
770 IF k=0 THEN RETURN
780 k=k-1:IF ABS(p(k))<10↑(-4) AND k>0 THEN 780
790 RETURN
800 '
810 ' Umbenennung
820 n=g(1):FOR i=0 TO n:p(i)=z(1,i):NEXT i
830 m=g(0):FOR i=0 TO m:q(i)=z(0,i):NEXT i:RETURN
840 '
850 ' GGT
860 GOSUB 820
870 IF n*m=0 THEN PRINT"Nicht mit diesem Programm":GOTO 30
880 l=(n-m+ABS(n-m))/2:IF n<m THEN k=n:GOTO 900
890 GOSUB 730
900 IF k=0 THEN 950
910 FOR i=0 TO k:r(i)=p(i):NEXT i
920 FOR i=0 TO m:p(i)=q(i):NEXT:n=m
930 FOR i=0 TO k:q(i)=r(i):NEXT:m=k
940 GOTO 880
950 PRINT:IF ABS(p(0))>10↑(-4) THEN PRINT"Der ggT ist 1":
    GOTO 30
960 PRINT"Der Grad des ggT ist"m
970 PRINT"Koeffizienten des ggT :"
980 FOR i=0 TO m:x=q(m-i)/q(m):GOSUB 1210:e(i)=x:PRINT x;:
    NEXT i
990 PRINT:IF o$="k" THEN RETURN
1000 GOTO 30
1010 '
1020 ' Multiplikation
1030 FOR j=0 TO n+m:x=0
1040 a=(j-m+ABS(j-m))/2:b=(j+n-ABS(j-n))/2
1050 FOR k=a TO b:x=x+p(k)*q(j-k):NEXT k
1060 r(j)=x:NEXT j:RETURN
1070 '
1080 ' Kuerzen
```

```
1090 GOSUB 860:n=g(1):FOR i=0 TO n:p(i)=z(1,i):NEXT i
1100 FOR i=0 TO m:q(i)=e(m-i):NEXT i
1110 GOSUB 720:PRINT:PRINT"Zaehler :"
1120 FOR i=1 TO 0 STEP -1:x=s(i):GOSUB 1210:e(1-i)=x
1130 PRINT x;:NEXT i:PRINT:g=1:GOSUB 160
1140 n=g(1):FOR i=0 TO n:p(i)=z(1,i):NEXT i
1150 GOSUB 720:PRINT:PRINT"Nenner :"
1160 FOR i=1 TO 0 STEP -1:x=s(i):GOSUB 1210:e(1-i)=x
1170 PRINT x;:NEXT i:g=1:GOSUB 160
1180 PRINT:GOTO 30
1190 '
1200 ' Runden
1210 IF x-INT(x)<10↑(-t) THEN x=INT(x):GOTO 1230
1220 IF INT(x)+1-x<10↑(-t) THEN x=INT(x)+1
1230 RETURN
```

Beispiel 1 : Man kürze und dividiere !

$$r(x) = \frac{2x^5 + 12x^4 + 9x^3 + 27x^2 - 423x + 189}{x^4 + 8x^3 + 12x^2 + 12x - 9}$$

Eingabe : 2 **RET** 12 **RET** 9 **RET** 27 **RET** -423 **RET** 189 **RET** p RET
1 **RET** 8 **RET** 12 **RET** 12 **RET** -9 **RET** p **RET** k **RET**

Ausgabe : Zähler $2x^3 + 15x - 63$

Nenner $x^2 + 2x + 3$

Eingabe : / **RET**

Ausgabe : $r(x) = 2x - 4 + \frac{17x - 51}{x^2 + 2x + 3}$

Beispiel 2 : Zusammenfassen !

$$r(x) = \frac{x + 2}{x^2 + x} + \frac{x - 3}{x^2 - 1}$$

Eingabe : 1 **RET** 2 **RET** p **RET** 1 **RET** 0 **RET** -1 **Ret** p **RET** * **RET**
1 **RET** -3 **RET** p **RET** 1 **RET** 1 **RET** 0 **RET** p **RET** * **RET**
+ **RET** 1 **RET** 1 **RET** 0 **RET** p **RET** 1 **RET** 0 **RET** -1 **RET**
p **RET** * **RET** k **RET**

Ergebnis : $r(x) = \frac{2x^2 - 2x - 2}{x^3 - x}$

Hornerschema

Nach Eingabe des Polynomgrads, der Koeffizienten $a_n,\ldots,a_0$ und der Entwicklungsstelle x berechnet das Programm die Zahlen des Hornerschemas (REP 51).

2. Hornerschema für Polynome mit reellen Koeffizienten

```
10 CLS:PRINT "Hornerschema":PRINT
20 INPUT "Polynomgrad ";n:PRINT
30 DIM a(n+1),b(n)
40 FOR k=n TO 0 STEP -1
50 PRINT"a("k")=";:INPUT a(k)
60 b(k)=a(k):NEXT k
70 PRINT:INPUT"x=";x:PRINT
80 FOR j=0 TO n:FOR i=0 TO n-j
90 a(n-i)=a(n-i)+a(n-i+1)*x
100 IF ABS(a(n-i))<10↑(-7) THEN a(n-i)=0
110 PRINT a(n-i);:NEXT i:PRINT:NEXT j
120 FOR k=0 TO n:a(k)=b(k):NEXT k:GOTO 70
```

Beispiel : $p(x) = x^4 - 5x^3 + 5x^2 + 5x - 6$ bei x=2

```
Ausgabe  :    1  -3  -1   3   0     =p(2)
              1  -1  -3  -3         =p'(2)
              1   1  -1             =p''(2)/2!
              1   3                 =p'''(2)/3!
              1                     =p''''(2)/4!
```

3. Hornerschema für Polynome mit komplexen Koeffizienten
(Koeffizienteneingabe : Re z,Im z)

```
10 CLS:PRINT "Hornerschema mit komplexen Koeffizienten"
20 PRINT:INPUT "Polynomgrad n=";n:PRINT
30 DIM c(n),im(n+1),r(n+1),b(n)
40 FOR k=n TO 0 STEP -1
50 PRINT"a("k")  : Re,Im";:INPUT r(k),im(k)
60 b(k)=r(k):c(k)=im(k)
70 NEXT k
80 PRINT:PRINT"z : Re,Im";:INPUT p,q:PRINT
90 PRINT "Ausgabeform (Re z/Im z)":PRINT
```

```
100 FOR j=0 TO n:FOR i=0 TO n-j
110 u=r(n-i+1):v=im(n-i+1)
120 x=p*u-q*v:y=u*q+v*p
130 r(n-i)=r(n-i)+x:im(n-i)=im(n-i)+y
140 PRINT "("r(n-i)"/"im(n-i)")";:NEXT i:PRINT:NEXT j
150 FOR k=0 TO n:r(k)=b(k):im(k)=c(k):NEXT k:GOTO 80
```

Beispiel : $p(x) = ix^3 + (1-i)x^2 + (2+i)x + 3+2i$, $x = 1+i$

Ausgabe : (0/1) (0/0) (2/1) (4/5)
(0/1) (-1/1) (0/1)
(0/1) (-2/2)
(0/1)

Aus diesem Schema kann man entnehmen :

1. Funktionswert an der Entwicklungsstelle (d.i. letzte Zahl der ersten Zeile) $p(1+i) = 4+5i$
2. Die Division von p(x) durch den Linearfaktor x - (1+i) ergibt als Polynomanteil (1. Zeile)
   $ix^2 + 2 + i$
   und als Rest $4+5i$.
   Somit ist
   $p(x) = (ix^2 + 2 + i)(x-1-i) + 4+5i$.
3. Die Umordnung von p(x) nach Potenzen des Linearfaktors z = x-(1+i) lautet :
   $p(x) = iz^3 + (-2+2i)z^2 + iz + 4+5i$
4. Die Ableitungen von p(x) bei x=1+i :
   $p(1+i) = 4+5i$ $p'(1+i) = i$ $p''(1+i) = 2(-2+2i)$
   $p'''(1+i) = 6i$

4 Interpolation

Zu n Punkten $(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)$ gibt es ein Polynom höchstens (n-1)ten Grades, das durch diese Punkte verläuft. Beim Newtonschen Interpolationsverfahren macht man den Ansatz

$$(*)\quad p(x) = a_0 + a_1(x-x_1) + a_2(x-x_1)(x-x_2) + \ldots + a_n(x-x_1)(x-x_2)\ldots(x-x_{n-1})$$

und bestimmt die 'Newton-Koeffizienten' a_k mit dem Verfahren der dividierten Differenzen.Durch Einfügen der Zeilen

```
145 PRINT a(0)
195 PRINT a(j)
```

kann man die Newton-Koeffizienten ausgeben lassen.
Nach Ausmultiplizieren von (*) ergibt sich :

$$p(x) = p_{n-1}x^{n-1} + p_{n-2}x^{n-2} + \ldots + p_1x + p_0$$

und das Programm berechnet die Koeffizienten $p_{n-1},..,p_1,p_0$.

```
10 CLS:PRINT"Interpolationsverfahren ";
20 PRINT"(Newton)":PRINT
30 INPUT "Zahl der Stuetzstellen";n
40 DIM c(n,n):c(1,1)=1
50 n=n-1:DIM p(n),x(n),y(n),a(n),d(n)
60 '
70 ' Eingabe der Stuetzstellen
80 PRINT"Eingabeform : x,y":PRINT
90 FOR k=0 TO n
100 PRINT k+1".Stuetzstelle";:INPUT x(k),y(k)
110 NEXT k
120 '
130 ' Dividierte Differenzen
140 a(0)=y(0):d(0)=y(0)
150 FOR j=1 TO n:d(j)=y(j)
160 FOR k=j-1 TO 0 STEP -1
170 d(k)=(d(k+1)-d(k))/(x(j)-x(k))
180 NEXT k
190 a(j)=d(0)
200 NEXT j:p(0)=a(0)
```

```
210 '
220 ' Koeff. des Interpol.polynoms
230 FOR i=2 TO n+1:FOR k=1 TO i
240 c(i,k)=c(i-1,k-1)-c(i-1,k)*x(i-2)
250 p(k-1)=p(k-1)+c(i,k)*a(i-1)
260 NEXT k:NEXT i
270 '
280 ' Ausgabe der Koeffizienten
290 PRINT:PRINT"Koeffizienten :":PRINT
300 FOR k=n TO 0 STEP -1
310 PRINT"p("k")="p(k):NEXT k
320 END
```

Beispiel

x	-2	-1	0	1	2	3
p(x)	-36	-7	-6	-9	56	429

Die Newton-Koeffizienten sind -36,29,-14,4,2,1 , so daß

$$p(x) = -36 +29(x+2) -14(x+2)(x+1) + 4(x+2)(x+1)x$$
$$+2(x+2)(x+1)x(x-1) + 1(x+2)(x+1)x(x-1)(x-2)$$

Umgeordnet ergibt sich

$$p(x) = x^5 + 2x^4 + 3x^3 - 4x^2 - 5x - 6.$$

Oft ist man nicht so sehr am Interpolationspolynom wie an dessen Funktionswerten interessiert.Diese lassen sich berechnen, auch ohne das Interpolationpolynom zu bestimmen

```
10 CLS:PRINT"Interpolationwert an der Stelle x":PRINT
20 INPUT "Zahl der Stuetzstellen";n
30 n=n-1:DIM x(n),y(n),d(n)
40 '
50 ' Eingabe der Stuetzstellen
60 PRINT"Eingabeform      : x,y":PRINT
70 FOR k=0 TO n
80 PRINT k+1".Stuetzstelle";:INPUT x(k),y(k)
90 NEXT k
100 '
110 PRINT:INPUT "x=";x
120 '
130 ' Berechnung von p(x)
```

```
140 d(0)=y(0)
150 FOR j=1 TO n:d(j)=y(j)
160 FOR k=j-1 TO 0 STEP -1
170 d(k)=d(k+1)+(x-x(j))*(d(k+1)-d(k))/(x(j)-x(k))
180 NEXT k:NEXT j
190 y=d(0)
200 PRINT:PRINT "p("x")="y
210 GOTO 110
```

Beispiel :

Berechnen Sie durch Interpolation p(0) und p(0.5)

x	-2	-1	1	2
p(x)	-19	-5	-1	1

Ergebnis : p(0) = -1 , p(0.5) = -0.875

Hinweis : In den beiden vorangehenden Programmen sind natürlich nur Stützstellen mit unterschiedlichen x-Werten zulässig , also x(i) ≠ x(j) für i≠j.

5 Polynomzerlegung

Jedes Polynom n-ten Grades zerfällt über dem Körper **C** der komplexen Zahlen in ein Produkt von n Linearfaktoren. Besitzt das Polynom reelle Koeffizienten, so treten komplexe Nullstellen stets paarweise (konjugiert komplex) auf.Hieraus ergibt sich, daß jedes Polynom mit reellen Koeffizienten über den reellen Zahlen in Linearfaktoren und ggf. noch unzerlegbare quadratische Faktoren zerfällt.

Für Polynome höchstens 5.Grades

$$p(x) = a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

berechnet dieses Programm die (ggf. komplexen) Nullstellen und bestimmt zu komplexen Nullstellen die zugehörigen unzerlegbaren quadratischen Faktoren.

```
10 CLS:GOSUB 1040:' Ueberschrift
20 '
30 ' Hauptprogramm
40 a=-1:c=-1:DIM z(3,1)
50 GOSUB 80:DIM d(n-1):DIM e(n+1):GOSUB 150:GOSUB 850:END
60 '
70 ' Koeffizienteneingeabe
80 PRINT"Polynomgrad ";:INPUT"n=";n:n0=n
90 IF(n<1 OR n>5) THEN 80
100 DIM l(n):PRINT"Koeffizienten :"
110 FOR k=n TO 0 STEP -1
120 PRINT"a("k")=";:INPUT l(k):NEXT k:PRINT:RETURN
130 '
140 ' Nullstellenermittlung
150 ON n GOTO 180,210,290,320,500
160 '
170 ' n=1
180 a=a+1:d(a)=-l(0)/l(1):RETURN
190 '
200 ' n=2
210 GOSUB 570:IF r=1 THEN 250
220 c=c+1:z(c,0)=s:z(c,1)=-v
230 c=c+1:z(c,0)=s:z(c,1)=v
```

```
240 p0=l(1):q0=l(0):RETURN
250 a=a+1:d(a)=x1
260 a=a+1:d(a)=x2:RETURN
270 '
280 ' n=3
290 GOSUB 760:a=a+1:d(a)=x:GOSUB 530:GOTO 210
300 '
310 ' n=4
320 DIM t(4):FOR k=0 TO 4
330 t(k)=l(k)/l(4):NEXT k
340 l(0)=t(1)*t(1)-t(0)*(4*t(2)-t(3)*t(3))
350 l(1)=t(1)*t(3)-4*t(0):l(2)=t(2)
360 l(4)=0:l(3)=1:n=3
370 GOSUB 760:GOSUB 530:GOSUB 570
380 IF r=0 THEN y=x:GOTO 400
390 y=x:IF x>x1 THEN y=x1
400 l(0)=t(0):l(1)=y:l(2)=1:n=2
410 GOSUB 570:q2=x1:q1=x2:IF w=0 THEN 440
420 p1=(t(3)*q1-t(1))/2/v
430 p2=(t(1)-t(3)*q2)/2/v:GOTO 460
440 l(1)=-t(3):l(0)=t(2)+y:GOSUB 570
450 p1=x1:p2=x2
460 l(1)=p1:l(0)=q1:GOSUB 210
470 l(1)=p2:l(0)=q2:GOSUB 210:RETURN
480 '
490 ' n=5
500 GOSUB 760:a=a+1:d(a)=x:GOSUB 530:GOTO 320
510 '
520 ' Abspalten eines Linearfaktors
530 GOSUB 710:FOR k=0 TO n-1
540 l(k)=e(k+1):NEXT k:n=n-1:RETURN
550 '
560 ' Quadratische Gleichung
570 s=-l(1)/l(2)/2
580 w=l(1)*l(1)-4*l(0)*l(2)
590 IF ABS(w)<10↑(-4) THEN w=0
```

```
600 v=SQR(ABS(w))/l(2)/2:IF w<0 THEN r=0:RETURN
610 x1=s-v:x2=s+v:r=1:RETURN
620 '
630 ' Funktionswert t(x)
640 h=0:ON n-1 GOTO 680,650,660,670
650 h=l(3):GOTO 680
660 h=l(3)+x*l(4):GOTO 680
670 h=l(3)+x*(l(4)+x*l(5))
680 f=l(0)+x*(l(1)+x*(l(2)+x*h)):RETURN
690 '
700 ' Hornerschema
710 e(n+1)=0:FOR k=0 TO n
720 e(n-k)=l(n-k)+e(n-k+1)*x:NEXT k:RETURN
730 '
740 ' Naeherungsverfahren fuer Nullstelle
750 ' in den Faellen n=3 und n=5
760 s=SGN(l(0)/l(n)):b=10:x=0
770 b=b/10:s=-s:IF ABS(b)<10↑(-7) THEN RETURN
780 x=x+b*s
790 GOSUB 640
800 IF f=0 THEN RETURN
810 IF l(n)*f*s<0 THEN GOTO 780
820 GOTO 770
830 '
840 ' Ausgabe der Nullstellen
850 IF a=-1 THEN 910
860 ' reelle Nullstellen
870 FOR k=0 TO a:x=d(k):GOSUB 990:d(k)=x
880 PRINT"x"k"="d(k):NEXT k:PRINT
890 IF c=-1 THEN RETURN
900 ' komplexe Nullstellen
910 FOR k=0 TO c:x=z(k,0):GOSUB 990:z(k,0)=x
920 PRINT"Re z"k"="z(k,0);
930 x=z(k,1):GOSUB 990:z(k,1)=x
940 PRINT"Im z"k"="z(k,1):NEXT k
950 GOSUB 1090
```

```
960 RETURN
970 '
980 ' Runden
990 x=1000*x:IF x-INT(x)<10↑(-4) THEN x=INT(x):GOTO 1010
1000 IF INT(x)+1-x<10↑(-4) THEN x=INT(x)+1
1010 x=x/1000:RETURN
1020 '
1030 ' Ueberschrift
1040 PRINT"Bestimmung der komplexen Nullstellen"
1050 PRINT"eines Polynoms hoechstens 5.Grades":PRINT:RETURN
1060 '
1070 ' Ausgabe der unzerlegbaren quadratischen
1080 ' Ausdruecke im Fall n>2
1090 IF n0<3 THEN RETURN
1100 PRINT:IF c=1 THEN 1150
1110 PRINT"Die zugehoerigen reellen quadratischen"
1120 PRINT"Ausdruecke lauten (normiert):"
1130 PRINT"x↑2+"p1"x+"q1:PRINT
1140 PRINT"x↑2+"p2"x+"q2:RETURN
1150 PRINT"Der zugehoerige reelle quadratische"
1160 PRINT "Ausdruck lautet (normiert):"
1170 x=p0:GOSUB 990:p=x:x=q0:GOSUB 990:q=x
1180 PRINT"x↑2+"p"x+"q:RETURN
```

Beispiel 1 $p(x) = x^4 + 6x^3 + 16x^2 + 26x + 15$

Nullstellen : $x_0 = -3$, $x_1 = -1$, $x_2 = -1-2i$, $x_3 = -1+2i$

Zerlegung : $p(x) = (x+3)(x+1)(x^2+2x+5)$.

Beispiel 2 $p(x) = x^4 + 6x^3 + 18x^2 + 30x + 25$

Nullstellen : $x_{0,1} = -1 \pm 2i$, $x_{2,3} = -2 \pm i$

Zerlegung : $p(x) = (x^2+2x+5)(x^2+4x+5)$.

Beispiel 3 $p(x) = 16x^5 - 48x^4 + 40x^3 - 12x^2 + 9x$

Nullstellen : $x_0 = 0$, $x_{1,2} = 1{,}5$, $x_{3,4} = \pm 0{,}5i$

Zerlegung : $p(x) = 16 \cdot x \cdot (x-1{,}5)^2 (x^2+0{,}25)$.

6 Ausgleichskurven

Zu n Punkten $(x_1,y_1),\ldots,(x_n,y_n)$ wird ein Polynom

$$p(x) = \sum_{k=0}^{m} p_k x^k \qquad (m<n-1)$$

derart bestimmt, daß

$$A = \sum_{i=1}^{n} (y_i - p(x_i))^2$$

minimal wird (Methode der kleinsten Quadrate).

Häufig tritt der Fall m=1 auf : Zu n Punkten ist die Regressionsgerade zu bestimmen.

Regressionsgerade

```
10 CLS:PRINT"Ausgleichgerade fuer n Punkte":PRINT
20 INPUT "Zahl der Punkte";n
30 n=n-1:DIM x(n),y(n)
40 '
50 ' Eingabe der Punkte
60 PRINT"Eingabeform    : x,y":PRINT
70 FOR k=0 TO n
80 PRINT k+1".Punkt";:INPUT x(k),y(k)
90 a=a+x(k):b=b+y(k)
100 c=c+x(k)*x(k):d=d+x(k)*y(k)
110 NEXT k
120 '
130 v=(c-a*a/(n+1))/n
140 k=(d-a*b/(n+1))/n
150 p=k/v:q=(b-p*a)/(n+1)
160 '
170 PRINT:PRINT "y="p"*x+"q
180 '
190 PRINT:PRINT"Berechnung von Funktionswerten (j/n)"
200 e$=INKEY$:IF e$="" THEN 200
210 IF e$="n" THEN 250
220 PRINT:INPUT "x=";x:PRINT
230 PRINT"y(";x;")=";p*x+q
240 GOTO 190
250 END
```

Das Programm läßt sich nach kleiner Abänderung auch verwenden, um folgende Kurventypen zum Ausgleich zu benutzen:

1a) $y = q*e^{px}$

Zeile 85 : y(k)=log(y(k))

Zeile 170 : Print"q="exp(q):Print"p="p:End

1b) $y = q + p*\ln(x)$

Zeile 85 : x(k)=log(x(k))

Zeile 170 : Print"q="q:Print"p="p:End

1c) $y = q*x^p$

Zeile 85 : x(k)=log(x(k)):y(k)=log(y(k))

Zeile 170 : Print"q="exp(q):Print"p="p:End

Beispiel 1 Ausgleich durch $y = px + q$

x	-1,2	2,3	2,6	5,1
y	-5,7	-0,6	-0,8	2,4

Ergebnis : $p \approx 1,289$, $q \approx -4,011$

Beispiel 2 Ausgleich durch $y = q*e^{px}$

x	0,4	0,8	1,5	2,3
y	0,14	0,17	0,22	0,30

Ergebnis : $q \approx 0,121$, $p \approx 0,395$

Beispiel 3 Ausgleich durch $y = q + p*\ln(x)$

x	2	3	4	5
y	3,35	3,95	4,4	4,7

Ergebnis : $q \approx 2,323$, $p \approx 1,484$

Beispiel 4 Ausgleich durch $y = q*x^p$

x	2	3	4	5
y	2,2	2,5	2,73	2,92

Ergebnis : $q \approx 1,777$, $p \approx 0,309$

Ausgleichspolynom

```
10 CLS:PRINT"Ausgleichspolynom fuer n Punkte":PRINT
20 INPUT "Zahl der Punkte";n
30 DIM x(n),z(n),c(n)
40 PRINT"Polynomgrad (<"n-1")";:INPUT m:DIM a(m+1,m+2)
50 '
60 ' Eingabe der Punkte
70 PRINT"Eingabeform       : x,y":PRINT
80 FOR k=1 TO n
90 PRINT k".Punkt";:INPUT x(k),z(k)
100 FOR i=1 TO m+1:FOR j=1 TO m+1
110 a(i,j)=a(i,j)+x(k)↑(i+j-2)
120 NEXT j
130 a(i,m+2)=a(i,m+2)+x(k)↑(i-1)*z(k)
140 NEXT i
150 NEXT k
160 '
170 n=m+1
```

```
180 FOR k=1 TO n
190 FOR i=1 TO n
200 IF ABS(a(i,k))<10↑(-5) THEN a(i,k)=0
210 IF (a(i,0)>0 OR a(i,k)=0) THEN NEXT i:GOTO 290
220 a(0,k)=i:a(i,0)=k
230 FOR l=1 TO n+1:c(l)=a(i,l):NEXT l
240 FOR j=1 TO n
250 y=a(j,k)/c(k):FOR l=1 TO n+1
260 a(j,l)=a(j,l)-c(l)*y
270 NEXT l:NEXT j
280 FOR l=1 TO n+1:a(i,l)=c(l)/c(k):NEXT l
290 NEXT k
300 FOR k=1 TO n
310 i=a(0,k)
320 x(k)=a(i,n+1)
330 NEXT k
340 '
350 PRINT:PRINT"Koeffizienten :":PRINT
360 FOR i=n TO 1 STEP-1
370 PRINT"p("i-1")=";x(i)
380 NEXT i
390 '
400 '
410 PRINT:PRINT"Berechnung von Polynomwerten (j/n)"
420 e$=INKEY$:IF e$="" THEN 420
430 IF e$="n" THEN END
440 PRINT:INPUT "x=";x
450 y=x(n):FOR k=n-1 TO 1 STEP-1:y=x(k)+x*y:NEXT k
460 PRINT:PRINT"p("x")=";y:GOTO 410
```

Beispiel : Zu gegebenen 6 Punkten ist ein Ausgleichspolynom 3. Grades zu bestimmen.

x	-1	-0,5	0	0,5	1	2
y	0,1	0,37	-0,05	-0,37	0,05	5,95

Ergebnis : $p(3) \approx 0{,}93$, $p(2) \approx 0{,}11$, $p(1) \approx -0{,}96$

$p(0) \approx -0{,}04$

Elementare Funktionen

Werden außer den bereits vom BASIC bereitgestellten Funktionen weitere Funktionen benötigt, so gibt es zwei Möglichkeiten.Entweder definiert man die Funktionen in einem Unterprogramm oder man benutzt den BASIC-Befehl DEF FN.

1.Definition elementarer Funktionen durch Unterprogramme*

```
10 CLS:PRINT"Elementare Funktionen y=f(x)"
20 DIM f$(13)
30 PRINT:PRINT"Berechnet werden die Funktionswerte von"
40 PRINT:PRINT:PRINT"1 Polynom":f$(1)="p("
50 PRINT"2 logb x (Logarithmus zur Basis b)":f$(2)="logb("
60 PRINT"3 sinh x":f$(3)="sinh("
70 PRINT"4 cosh x":f$(4)="cosh("
80 PRINT"5 tanh x":f$(5)="tanh("
90 PRINT"6 coth x":f$(6)="coth("
100 PRINT"7 Arsinh x":f$(7)="Arsinh("
110 PRINT"8 Arcosh x":f$(8)="Arcosh("
120 PRINT"9 Artanh x":f$(9)="Artanh("
130 PRINT"10 Arcoth x":f$(10)="Arcoth("
140 PRINT"11 Arcsin x":f$(11)="Arcsin("
150 PRINT"12 Arccos x":f$(12)="Arccos("
160 PRINT"13 Arccot x":f$(13)="Arccot("
170 '
180 PRINT:INPUT"Welche Funktion";e
190 CLS:IF e>1 THEN 340
200 '
210 PRINT"Polynomwert p(x)":PRINT
220 INPUT "Polynomgrad";n:PRINT
230 PRINT"Koeffizienteneingabe :"
240 FOR k=n TO 0 STEP-1
250 PRINT"p("k")=";:INPUT a(k):NEXT k
260 PRINT:INPUT "x=";x
270 y=a(n):FOR k=n-1 TO 0 STEP-1:y=a(k)+x*y:NEXT k
280 PRINT:PRINT f$(1)x")="y:PRINT
290 PRINT "Weiterer Wert dieses Polynoms (j/n)"
```

*) gegebenenfalls PI = 4*ATN(1) definieren!

```
300 e$=INKEY$:IF e$="" THEN 300
310 IF e$="n" THEN CLS:GOTO 30
320 GOTO 260
330 '
340 PRINT f$(e)"x)":PRINT:INPUT "x=";x
350 IF e>2 AND e<7 THEN GOSUB 470
360 ON e-1 GOSUB 440,480,490,500,510,540,550,570,590,620,
    650,680
370 IF i=1 THEN i=0:GOTO 30
380 '
390 PRINT:PRINT f$(e)x")="y
400 IF e>10 THEN PRINT "im Gradmass : "y*180/PI" GRAD"
410 e$=INKEY$:IF e$="" THEN 410
420 CLS:GOTO 30
430 '
440 INPUT"Basis b";b:y=LOG(x)/LOG(b):RETURN
450 '
460 ' Hyperbelfunktionen
470 u=EXP(x):v=1/u:RETURN
480 y=(u-v)/2:RETURN
490 y=(u+v)/2:RETURN
500 y=(u-v)/(u+v):RETURN
510 IF x=0 THEN 690:RETURN
520 y=(u+v)/(u-v):RETURN
530 ' Umkehrfktn der Hyperbelfunktionen
540 y=LOG(x+SQR(x*x+1)):RETURN
550 IF x<1 THEN 690
560 y=LOG(x+SQR(x*x-1)):RETURN
570 IF ABS(x)>=1 THEN 690
580 y=LOG((1+x)/(1-x))/2:RETURN
590 IF ABS(x)<=1 THEN 690
600 y=LOG((x+1)/(x-1))/2:RETURN
610 ' Arcus-Funktionen
620 IF ABS(x)>1 THEN 690
630 y=SGN(x)*PI/2:IF ABS(x)<1 THEN y=ATN(x/SQR(1-x*x))
640 RETURN
```

```
650 IF ABS(x)>1 THEN 690
660 y=PI/2:IF x<>0 THEN y=ATN(SQR(1-x*x)/x)+PI*(1-SGN(x))/2
670 RETURN
680 y=PI/2-ATN(x):RETURN
690 PRINT:PRINT"Definitionsbereich beachten":i=1:RETURN
```

Beispiele zu Polynomwerten :

$p(x) = 4.5x^4 - 3x^2 + 5.6x - 7$

$p(1.2) = 4.7310001 \quad p(-2.1) = 55.52645$

Beispiele zum Logarithmus :

x=125 , b=5 : logb(125)= 3

x= 32 , b=0.5 : logb(32)= -5

x=20 , b=10 : logb(20)=1.30103

Beispiele zu den Hyperbelfunktionen :

sinh(1.23) = 1.56446848 , cosh(1.23) = 1.85676106

tanh(1.23) = 0.842579326 , coth(1.23) = 1.18683187

Beispiele zu den Area-Funktionen :

Arsinh(2.3) = 1.57027854 , Arcosh(2.3) = 1.47504478

Artanh(2.3) : nicht def. , Arcoth(2.3) = 0.465779102

Beispiele zu den Arcus-Funktionen :

Arcsin(0.7) = 0.775397497 ≙ 44.4°

Arccos(0.7) = 0.79539883 ≙ 45.6°

Arccot(0.7) = 0.960070362 ≙ 55.0°

Die Verwendung von Unterprogrammen hat den Vorteil, daß abschnittsweise definierte Funktionen durch Fallunterscheidung leicht zu definieren sind. Bei Verwendung von DEF FN muß man in diesem Fall mehr Mühe aufwenden :
Um Arcsin x zu definieren , verwendet man die Beziehung :

$$\text{Arcsin } x = \text{Arctan}\left(x \Big/ \sqrt{1-x^2}\right) \quad \text{für } |x|<1 .$$

Nun ist der Arcsin aber auch für $|x|=1$ definiert und dort erwartet man : $\text{Arcsin}(-1)=-\pi/2$ bzw. $\text{Arcsin}(1)=\pi/2$.
Um diese Fallunterscheidungen in einer Zeile unterzubrin-

gen, benutzt man die 'Vergleichsoperatoren'. Ausdrücke wie 'a=b' , 'a<b' oder 'a<=b' bekommen vom BASIC-Interpreter den Wert -1,wenn sie wahr sind bzw. 0 ,wenn sie falsch sind. Und damit klappt es : s. Arcsin-Definition im folgenden Programm.

Der DEF FN-Befehl läßt mehrere Veränderliche zu (zumindest im Schneider-BASIC und GWBASIC) :s.Logarithmus-Def.

2. Definition elementarer Funktionen mit DEF FN *:

```
10 PRINT"Definiert sind :
20 PRINT"fnlg(b,x)"
30 PRINT"fncot"
40 PRINT"fnarcsin , fnarccos , fnarccot"
50 PRINT"fnsinh , fncosh , fntanh , fncoth"
60 PRINT"fnarsinh , fnarcosh , fnartanh , fnarcoth"
70 DEF FNlg(b,x)=LOG(x)/LOG(b)
80 DEF FNcot(x)=1/TAN(x)
90 DEF FNarcsin(x)=-PI*(x*x=1)*SGN(x)/2-
   ATN(x/SQR(1-x*x-(x*x=1)))*(x*x<1)
100 DEF FNarccos(x)=PI/2-FNarcsin(x)
110 DEF FNarccot(x)=PI/2-ATN(x)
120 DEF FNsinh(x)=(EXP(x)-EXP(-x))/2
130 DEF FNcosh(x)=(EXP(x)+EXP(-x))/2
140 DEF FNtanh(x)=FNsinh(x)/FNcoshh(x)
150 DEF FNcoth(x)=1/FNtanh(x)
160 DEF FNarsinh(x)=LOG(x+SQR(x*x+1))
170 DEF FNarcosh(x)=LOG(x+SQR(x*x-1))
180 DEF FNartanh(x)=(LOG((1+x)/(1-x)))/2
190 DEF FNarcoth(x)=(LOG((x+1)/(x-1)))/2
```

Die Zeilen 10-60 dienen natürlich nur der Information.

Beispiele :

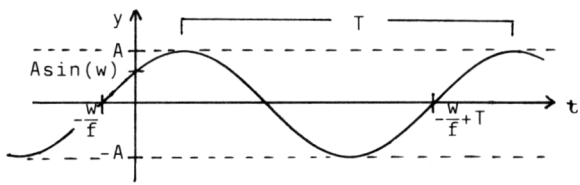
```
? fnlg(2,1024) : 10
? fncot(1.5) : 7.09148E-02
? fnarcsin(0.5) : 0.523598776 , ? fnarcsin(1) : 1.57079633
? fnarccos(0) : 1.57079633
? fnartanh(0.5) : 0.549306145
```

*) gegebenenfalls PI = 4*ATN(1) definieren!

Schwingungen gleicher Frequenz

$$y = A\sin(ft + w)$$

heißt harmonische oder Sinusschwingung (**REP** 382).



Die charakteristischen Größen einer Schwingung:

A : Amplitude (halber Wert der Schwingungsweite)

f : Kreisfrequenz (Zahl der Schwingungen in 2π Sek.)

w : Phasenverschiebung (Anfangsphase)

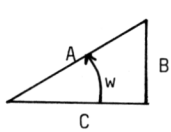
$T = \frac{2\pi}{f}$: Periode (Schwingungsdauer)

$\frac{1}{T} = \frac{f}{2\pi}$: Frequenz (Zahl der Schwingungen in 1 Sekunde)

Bemerkung: Cosinusschwingungen lassen sich leicht in Sinusschwingungen umschreiben:

$$A\cos(ft+w) = A\sin(ft+w+\tfrac{\pi}{2})$$

Zerlegung: Eine Sinusschwingung mit beliebigem Phasenwinkel läßt sich als Summe einer reinen Cosinusschwingung und einer reinen Sinusschwingung darstellen:



$$A\sin(ft+w) = B\cos(ft) + C\sin(ft)$$

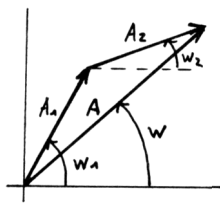
$A = \sqrt{B^2 + C^2}$ | $B = A\sin(w)$

$\tan(w) = \frac{B}{C}$ | $C = A\cos(w)$

(Quadr. beachten!)

Diese Zusammenhänge ergeben sich aus dem Additionstheorem für die Sinus-Funktion (**REP** 61).

Überlagerung: Die Addition (Überlagerung) zweier Schwingungen derselben Frequenz ergibt wieder eine Schwingung derselben Frequenz, jedoch mit entsprechend geänderter Amplitude und Phase:



$$A_1\sin(ft+w_1) + A_2\sin(ft+w_2) = A\sin(ft+w)$$

$$A = \sqrt{A_1^2 + A_2^2 + 2A_1A_2\cos(w_1-w_2)}$$

$$\tan(w) = \frac{A_1\sin(w_1) + A_2\sin(w_2)}{A_1\cos(w_1) + A_2\cos(w_2)}$$

(Quadranten beachten!)

Zeigerdiagramm: Vektoraddition zweier Schwingungen gleicher Frequenz

Wählt man bei folgendem Programm das Unterprogramm "1" (**Zerlegung**), so liefert es bei gegebenen A,w die Werte B,C. w kann nach Wahl in Bogenmaß (RAD) "0" oder in Grad (DEG) "1" eingegeben werden.

Wählt man das Unterprogramm "2" (**Überlagerung**), so liefert es bei gegebenen A_1,w_1,A_2,w_2 die Werte A,w.
Sowohl bei der Eingabe als auch bei der Ausgabe von w kann zwischen Bogenmaß (RAD) "0" und Grad (DEG) "1" gewählt werden.

1 Zerlegung und Überlagerung von Schwingungen (SHARP)

Das folgende Programm ist für den SHARP gedacht (für andere Rechner siehe das nächste Programm).

```
10 PAUSE"Schwingungen"
20 PAUSE"1:Zerlegung"
30 PAUSE"2:Ueberlagerung"
40 INPUT "Prog 1 oder 2?",p
50 ON p GOTO 60,160
60 PAUSE"Zerlegung"
70 INPUT "RAD 0 / DEG 1?",m
80 INPUT "A=",a
```

```
90 INPUT "W=",w
100 GOSUB 410
110 z=a*COS(w):GOSUB 360:c=z
120 z=a*SIN(w):GOSUB 360:b=z
130 PRINT"b=",b
140 PRINT"c=",c
150 END
160 PAUSE"Ueberlagerung"
170 INPUT "Ein:RAD 0 / DEG 1?",m
180 INPUT "Aus:RAD 0 / DEG 1?",n
190 INPUT "A1=",b
200 INPUT "W1=",w:GOSUB 410:u=w
210 INPUT "A2=",c
220 INPUT "W2=",w:GOSUB 410:v=w
230 z=SQR(b*b+c*c+2*b*c*COS(u-v))
240 GOSUB 360:a=z
250 IF a=0 THEN 320
260 y=b*SIN(u)+c*SIN(v)
270 x=b*COS(u)+c*COS(v)
280 IF ABS(x)<0.000001 THEN LET w=90*SGN(y):
    m=2-n:GOSUB 410:GOTO 320
290 w=ATN(y/x)-180*(x<0)
300 m=2-n:GOSUB 410
310 z=w: GOSUB 360:w=z
320 PRINT"A=",a
330 IF a=0 THEN 350
340 PRINT"W=",w
350 END
360 z=z*1000:i=INT(z)
370 IF z-i<0.001 THEN LET z=i:GOTO 390
380 IF i+1-z<0.001 THEN LET z=i+1
390 z=z/1000
400 RETURN
410 w=w*(180/PI)↑(1-m)
420 RETURN
```

Beispiel 1: Man zerlege $-2\sin(3t+1)$

RUN
Schwingungen
1: Zerlegung
2: Überlagerung

Prog 1 oder 2 ? 1
Zerlegung
RAD 0 / DEG 1 ? 0

A = -2
w = 1 ergibt

B = -1.68294
C = -1.08060

Also $-2\sin(3t+1) = -1{,}683\cos(3t) - 1{,}081\sin(3t)$

Beispiel 2: Man zerlege $2\sin(2t-90°)$

Prog 1 oder 2 ? 1
RAD 0 / DEG 1 ? 1

A = 2
w = -90 ergibt

B = -2
C = 0

Also $2\sin(2t-90°) = -2\cos(2t)$

Beispiel 3: Man überlagere: $3\sin(t+1) - 3\cos(t+1+\frac{\pi}{2})$

Aus $\cos(\beta)=\sin(\beta+\frac{\pi}{2})$ folgt: $3\cos(t+1+\frac{\pi}{2}) = 3\sin(t+1+\pi)$

Prog 1 oder 2 ? 2
EIN: RAD 0 / DEG 1 ? 0
AUS: RAD 0 / DEG 1 ? 0

$A_1 = 3$
$w_1 = 1$
$A_2 = -3$
$w_2 = 1+\pi$ ergibt

A = 6
w = 1 .Also $3\sin(t+1) - 3\cos(t+1+\frac{\pi}{2}) = 6\sin(t+1)$.

Beispiel 4: Man überlagere: $2\sin(2t+120°) - 3\sin(2t-60°)$

Wir wollen dieses Beispiel in allen Möglichkeiten der Ein- und Ausgabe durchprobieren:

Umrechnung von Grad in Bogenmaß:

$120° \mathrel{\hat{=}} 120/180*\pi = 2.094395$
$-60° \mathrel{\hat{=}} -60/180*\pi = -1.047198$ und $-240° \mathrel{\hat{=}} -4.188790$

EIN	AUS					
1	1	$2\sin(2t+120°)$	$- 3\sin(2t-60°)$	$=$	$5\sin(2t-240°)$
1	0	$2\sin(2t+120°)$	$- 3\sin(2t-60°)$	$=$	$5\sin(2t-4,189)$
0	1	$2\sin(2t+2,094)$	$- 3\sin(2t-1,047)$	$=$	$5\sin(2t-240°)$
0	0	$2\sin(2t+2,094)$	$- 3\sin(2t-1,047)$	$=$	$5\sin(2t-4,189)$

Beispiel 5: Man überlagere: $2\sin(2t+\frac{2}{3}\pi) - 3\sin(2t-\frac{1}{3}\pi)$ und zerlege die Überlagerung in Cosinus- und Sinusschwingung.

Wir möchten nach dem Überlagern noch Zerlegen und fügen folglich folgende Zeile in das Programm ein:

```
315 m=n:GOTO 100
```

Nun starten wir das Programm:

```
Prog 1 oder 2 ?        2
EIN: RAD 0 / DEG 1 ?   0
AUS: RAD 0 / DEG 1 ?   1
```

$A_1 = 2$
$w_1 = 2/3*\pi$
$A_2 = -3$
$w_2 = -\pi/3$ ergibt

$A = 4.33012$
$B = -2.5$

Also $2\sin(2t+\frac{2}{3}\pi) - 3\sin(2t-\frac{1}{3}\pi) = 4,330\cos(2t) - 2,5\sin(2t)$.

2 Zerlegung und Überlagerung von Schwingungen *)

```
10 REM ***** Auswahl *****
20 CLS:PRINT "    Zerlegung und Ueberlagerung von"
30 PRINT: PRINT "             Schwingungen":PRINT
40 PRINT "        (Bogenmass oder Grad)"
50 PRINT:PRINT:PRINT
60 PRINT "Zerlegung:":PRINT
70 PRINT"A*sin(f*t+w) = B*cos(f*t) + C*sin(f*t)":PRINT
80 PRINT: PRINT "Ueberlagerung:":PRINT
90 PRINT"A1*sin(f*t+w1) + A2*sin(f*t+w2)"
100 PRINT:PRINT"               =":PRINT
110 PRINT"          A*sin(f*t+w)"
120 PRINT:PRINT:PRINT:PRINT
130 INPUT "Zerlegung 1 oder Ueberlagerung 2";p
140 CLS:ON p GOTO 170,290
150 REM*************************************
160 REM ***** Zerlegung *****
170 PRINT"Zerlegung:":PRINT
180 PRINT"A*sin(f*t+w) = B*cos(f*t) + C*sin(f*t)":PRINT
190 PRINT: PRINT"Eingabe von w in":PRINT
200 INPUT "Bogenmass 0 oder Grad 1";m
210 PRINT:PRINT
220 INPUT "A=";a:INPUT "w=";w:GOSUB 620
230 z=a*COS(w):GOSUB 560:c=z:PRINT
240 z=a*SIN(w):GOSUB 560:b=z
250 PRINT: PRINT "B=",b
260 PRINT"C=",c:PRINT
270 END
280 REM ***** Ueberlagerung *****
290 PRINT"Ueberlagerung:":PRINT
300 PRINT"A1*sin(f*t+w1) + A2*sin(f*t+w2)"
310 PRINT:PRINT"  = A*sin(f*t+w)":PRINT:PRINT
320 INPUT "Eingabe in Bogenmass 0 oder Grad 1";m
```

*) gegebenenfalls Pi=4*ATN(1) definieren!

```
330 PRINT
340 INPUT "Ausgabe in Bogenmass 0 oder Grad 1";n
350 PRINT:PRINT
360 INPUT "A1=";b:INPUT "w1=";w
370 GOSUB 620:u=w
380 PRINT
390 INPUT "A2=";c:INPUT "w2=";w
400 GOSUB 620:v=w
410 z=SQR(b*b+c*c+2*b*c*COS(u-v))
420 GOSUB 560:a=z
430 IF a=0 THEN 500
440 y=b*SIN(u)+c*SIN(v)
450 x=b*COS(u)+c*COS(v)
460 IF ABS(x)<0.000001 THEN LET w=PI*SGN(y)/2:
    m=-n:GOSUB 620:GOTO 500
470 w=ATN(y/x)-PI*(x<0)
480 m=-n:GOSUB 620
490 z=w: GOSUB 560:w=z
500 PRINT:PRINT:PRINT"A=",a
510 IF a=0 THEN 530
520 PRINT"w=",w:PRINT
530 END
540 REM**************************************
550 REM ***** Runden *****
560 z=z*1000:i=INT(z)
570 IF z-i<0.001 THEN LET z=i:GOTO 590
580 IF i+1-z<0.001 THEN LET z=i+1
590 z=z/1000
600 RETURN
610 REM ***** Bogenmass / Grad *****
620 w=w*(PI/180)↑m
630 RETURN
```

Beispiel 1: Man zerlege die Schwingung $3\sin(2t+\frac{\pi}{4})$ in eine reine Cosinus- und eine reine Sinusschwingung.

Zerlegung 1 oder Überlagerung 2? 1
Eingabe von w in Bogenmaß 0 oder Grad 1? 0

A = 3
w = 0.785398163 ergibt

B = 2.12132034
C = 2.12132035

Also: $3\sin(2t+\frac{\pi}{4}) = 2.121\cos(2t) + 2.121\sin(2t)$

Beispiel 2: Man zerlege $2\sin(\frac{t}{3}-90°)$.

Zerlegung 1 oder Überlagerung 2? 1
Eingabe von w in Bogenmaß 0 oder Grad 1? 1

A = 2
w = -90 ergibt

B = -2
C = 0

Also $2\sin(\frac{t}{3}-90°) = -2\cos(\frac{t}{3})$

Beispiel 3: Man überlagere: $-2\sin(3t+2) + 0{,}5\sin(3t-1{,}3)$

Zerlegung 1 oder Überlagerung 2? 2
Eingabe in Bogenmaß 0 oder Grad 1? 0
Ausgabe in Bogenmaß 0 oder Grad 1? 0

A_1 = -2
w_1 = 2

A_2 = 0.5
w_2 = -1.3 ergibt

A = 2.49498688
w = -1.17321045

Also $-2\sin(3t+2) + 0{,}5\sin(3t-1{,}3) = 2{,}495\sin(3t-1{,}173)$

Beispiel 4: Man überlagere: $3\sin(4t+1) - 3\cos(4t+1+\frac{\pi}{2})$ und gebe die Phase des Ergebnisses in Grad an.

Es ist $\cos(\beta) = \sin(\beta+\frac{\pi}{2})$, also $-3\cos(t+1+\frac{\pi}{2}) = -3\sin(t+1+\pi)$.

Zerlegung 1 oder Überlagerung 2? 2
Eingabe in Bogenmaß 0 oder Grad 1? 0
Ausgabe in Bogenmaß 0 oder Grad 1? 1

$A_1 = 3$
$w_1 = 1$

$A_2 = -3$
$w_2 = 4.14159265$ ergibt

$A = 6$
$w = 57.2957794$

Also $3\sin(4t+1) - 3\cos(4t+1+\frac{\pi}{2}) = 6\sin(4t+57{,}296°)$

Es ist $\text{arc}(57{,}2957794°) = 57{,}2957794 * \pi/180 = 1$.
Also ist $6\sin(4t+57{,}296°) = 6\sin(4t+1)$. Das sieht man auch daran, daß $-3\cos(t+1+\frac{\pi}{2}) = 3\cos(t+1-\frac{\pi}{2}) = 3\sin(t+1)$ ist.

Beispiel 5: Man überlagere: $-2\sin(3t+60°) + 3\sin(3t-45°)$

Zerlegung 1 oder Überlagerung 2? 2
Eingabe in Bogenmaß 0 oder Grad 1? 1
Ausgabe in Bogenmaß 0 oder Grad 1? 1

$A_1 = -2$
$w_1 = 60$

$A_2 = 3$
$w_2 = -45$ ergibt

$A = 4.01320677$
$w = -73.7751503$

Also $-2\sin(3t+60°) + 3\sin(3t-45°) = 4{,}013\sin(3t-73{,}775°)$

3 Überlagerung von Schwingungen (Bildschirm)

Folgendes Programm führt wie die beiden vorigen Programme die Überlagerung von Sinusschwingungen derselben Frequenz durch.
Die beiden Schwingungen sowie die Überlagerung werden in einem geeigneten Koordinatensystem am Bildschirm ausgegeben.

Die charakteristischen Größen der Überlagerung werden in Bogenmaß (RAD) und Grad (DEG) angegeben.

Das Programm läuft auf IBM-kompatiblen PCs (1) und auf dem Schneider (0). *)

```
10 REM ***** Eingabe *****
20 CLS
30 PRINT"falls der Computer IBM-kompatibel ist 1,"
40 INPUT "bei Schneider CPC 0 eingeben";ir
50 CLS:IF ir=0 THEN MODE 1:GOTO 70
60 PI=4*ATN(1):screen 0
70 PRINT"   Ueberlagerung von Schwingungen   "
80 PRINT:PRINT
90 PRINT"         a * sin (f*t + w)"
100 PRINT:PRINT:PRINT
110 INPUT "Bogenmass 0 oder Grad 1";m
120 PRINT:PRINT
130 PRINT"erste Schwingung:"
140 PRINT
150 INPUT "a=";a(1)
160 INPUT "w=";w(1)
170 PRINT
180 PRINT"zweite Schwingung:"
190 PRINT
200 INPUT "a=";a(2)
210 INPUT "w=";w(2)
220 PRINT
230 INPUT "Frequenz f =";f
240 IF f=0 THEN 230
```

*) gegebenenfalls Pi=4*ATN(1) definieren!

```
250 REM ***** positives f *****
260 IF f<0 THEN LET f=-f:a(1)=-a(1):a(2)=-a(2):
    w(1)=-w(1):w(2)=-w(2)
270 REM**** neue Amplitude, Phase ****
280 s=PI/180:IF m=0 THEN 300
290 w(1)=s*w(1):w(2)=s*w(2)
300 b=a(1):u=w(1):c=a(2):v=w(2)
310 z=SQR(B*B+C*C+2*B*C*COS(U-V))
320 GOSUB 460:a(3)=z
330 z=B*SIN(U)+C*SIN(V):GOSUB 460:y=z
340 z=B*COS(U)+C*COS(V):GOSUB 460:x=z
350 IF y=0 THEN LET w(3)=0:GOTO 390
360 IF x=0 THEN LET w(3)=PI*SGN(y)/2:GOTO 390
370 z=ATN(y/x)-PI*(x<0)*SGN(y)
380 GOSUB 460:w(3)=z
390 z=-w(3)/f:GOSUB 460:w0=z
400 z=w0/s:GOSUB 460:w1=z
410 z=2*PI/f:GOSUB 460:p0=z
420 z=360/f:GOSUB 460:p1=z
430 z=a(3)*SIN(w(3)):GOSUB 460:y0=z
440 GOTO 530
450 REM ***** Runden *****
460 z=z*1000:i=INT(z)
470 IF z-i<0.001 THEN LET z=i:GOTO 490
480 IF i+1-z<0.001 THEN LET z=i+1
490 z=z/1000
500 RETURN
510 REM**********************
520 REM**** Skizze ****
530 CLS:IF ir=1 THEN screen 2
540 REM**** Achsenkreuz ****
550 am=ABS(a(1))
560 IF am<ABS(a(2)) THEN LET am=ABS(a(2))
570 IF am<ABS(a(3)) THEN LET am=ABS(a(3))
580 x1=0:y1=200:x2=640:y2=200:GOSUB 1070
590 x1=320:y1=115:x2=320:y2=300:GOSUB 1070
600 e=10↑INT(LOG10(p0))
610 q=25*e*f
620 p=320 MOD q
```

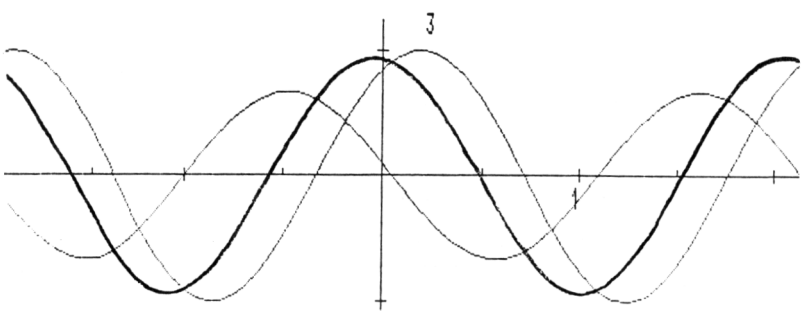
```
630 s=INT(320/q)
640 FOR i=1 TO s
650 FOR j=-1 TO 1 STEP 2
660 x1=320+i*j*q:y1=200+4*(i/2=INT(i/2))
670 x2=x1:y2=204:GOSUB 1070
680 NEXT j
690 NEXT i
700 zei=14:sp=INT(20+3*f*e+(e<>1)):GOSUB 1100:PRINT e
710 FOR i=0 TO 1
720 x1=316:y1=120+i*160
730 x2=324:y2=120+i*160:GOSUB 1070
740 NEXT i:IF am=0 THEN 770
750 zei=7:sp=22:GOSUB 1100:PRINT am
760 REM**** Kurven ****
770 FOR j=1 TO 3
780 IF ir=0 THEN GRAPHICS PEN 2:INK 2,7
790 IF a(1)*a(2)=0 THEN LET j=3
800 a=a(j):w=w(j)
810 IF ir=0 AND j=3 THEN GRAPHICS PEN 1
820 IF am=0 THEN 920
830 FOR i=0 TO 640 STEP 4
840 x2=i:y2=200+a*SIN((i-320)/50+w)*80/am
850 IF i=0 THEN x1=x2:y1=y2:GOTO 870
860 GOSUB 1070:x1=x2:y1=y2
870 NEXT i
880 NEXT j
890 IF ir=0 THEN GRAPHICS PEN 1
900 REM**********************
910 REM**** Ausgabe ****
920 zei=5:sp=1:GOSUB 1100
930 w$=STR$(-f*w0)+")"
940 IF w0<=0 THEN LET w$="+"+w$
950 PRINT a(3);"sin(";f;"t";w$
960 zei=19:sp=1:GOSUB 1100
970 PRINT:PRINT"Amplitude          = ";a(3)
980 IF a(3)=0 THEN 1060
990 PRINT"Kreisfrequenz      = ";f
1000 PRINT"Schnittp y-Achse   = ";y0
1010 PRINT
```

```
1020 PRINT"Periode(Bogenmass) = ";p0
1030 PRINT"Periode(Grad)      = ";p1
1040 PRINT"Nullst(Bogenmass)  = ";w0
1050 PRINT"Nullst(Grad)       = ";w1
1060 END
1070 IF ir=1 THEN 1090
1080 PLOT x1,y1:DRAW x2,y2:RETURN
1090 LINE(x1,-y1/2+200)-(x2,-y2/2+200):RETURN
1100 IF ir=1 THEN 1120
1110 LOCATE sp,zei:RETURN
1120 LOCATE zei,2*sp:RETURN
```

Beispiel 6: Man überlagere: $2\sin(3t+3) + 3\sin(3t+1)$

Nach Eingabe der Kennziffer 0 oder 1 starten wir das Programm erneut.
Wir wählen 0 für Bogenmaß, geben die Parameter für die beiden Schwingungen ein und erhalten:

2.82953 sin(3t + 1.698043)




```
Amplitude          = 2.82953
Kreisfrequenz      = 3
Schnittp y-Achse   = 2.806653

Periode (Bogenmaß) = 2.094395
Periode (Grad)     = 120
Nullst (Bogenmaß)  = -0.5660145
Nullst (Grad)      = -32.43024
```

1 Integration (Gauß)

Folgendes Programm berechnet in n Schritten näherungsweise das bestimmte Integral $I = \int_a^b f(x)dx$ einer Funktion f(x) in den Grenzen von a bis b.

Die Funktion f(x) ist in Zeile 110 zu programmieren.

Das Gauß-Verfahren ist schnell und genau, man kommt mit kleinen Schrittzahlen aus.

```
10 INPUT "Integrationsgrenzen a,b";a,b
20 PRINT: INPUT "Schrittzahl n";n
30 h=(b-a)/2/n:g=0:d=SQR(0.6)*h:z=a-h
40 FOR i=1 TO n:z=z+2*h
50 FOR j=-1 TO 1:x=z+j*d:GOSUB 110
60 g=g+(8-j*j*3)*f
70 NEXT j:NEXT i
80 g=g*h/9
90 PRINT "I=";g
100 END
110 f=
120 RETURN
```

Beispiel 1: Bogenlänge der Sinuskurve $f(x) = \sin(x)$ für $0 \leq x \leq \frac{\pi}{2}$

Zur Erinnerung (siehe etwa **BIN** 43):

(1) Kurve in Parameterform : $\vec{x}(t) = (x(t), y(t))$, $a \leq t \leq b$

Bogenlänge $L = \int_a^b \sqrt{\dot{x}^2(t) + \dot{y}^2(t)}\, dt$

(2) Kurve als Graph einer Funktion : $y = f(x)$, $a \leq x \leq b$

Bogenlänge $L = \int_a^b \sqrt{1 + f'^2(x)}\, dx$

(3) Kurve in Polarkoordinaten : $r = r(\varphi)$, $a \leq \varphi \leq b$

Bogenlänge $L = \int_a^b \sqrt{r^2(\varphi) + \dot{r}^2(\varphi)}\, d\varphi$

Bogenlängen sind wegen der unter dem Integralzeichen auftretenden Wurzeln oft schwierig und manchmal elementar gar nicht zu berechnen.

Zu berechnen ist also: $L = \int_0^{\pi/2} \sqrt{1+f'^2(x)}\,dx = \int_0^{\pi/2} \sqrt{1+\cos^2 x}\,dx$.

Dieses Integral ist elementar unlösbar, d.h es gibt keine elementare Funktion, deren Ableitung $\sqrt{1+\cos^2(x)}$ ist.

Wir programmieren:

```
110 f=SQR(1+cos(x)*cos(x))
```

und starten das Programm:

```
a,b=0,1.570796327
Schrittzahl n=1     ergibt      I = 1.91034403
desgleichen mit
Schrittzahl n=2     ergibt      I = 1.910101173
            n=3     ergibt      I = 1.91009894
            n=10    ergibt      I = 1.91009889
```

Beispiel 2: (siehe **REP** 159 (46)) $\int_0^{\pi/2} \frac{\cos^3 x}{1-\sin x}\,dx$

```
110 f=cos(x)*cos(x)*cos(x)/(1-sin(x))
```

```
a,b = 0,1.570796327
Schrittzahl n = 1              ergibt      I = 1.50035535
Schrittzahl n = 3              ergibt      I = 1.50000035
                               Der exakte Wert ist  1,5
```

Beispiel 3: Fläche zwischen x-Achse und Funktion $f(x)=e^{-x^2}$ für $0<x<1$

Zu berechnen ist $F = \int_0^1 e^{-x^2}\,dx$

Auch dieses Integral ist elementar unlösbar!

```
110 f=exp(-x*x)
```

```
a,b = 0,1
Schrittzahl n = 1              ergibt      I = 0.746814584
            n = 2              ergibt      I = 0.746824097
            n = 10             ergibt      I = 0.746824133
```

Bei diesen Beispielen fällt auf, daß man schon mit kleinen Schrittzahlen gute Näherungen bekommt.
Das Gauß-Verfahren ist wesentlich leistungsfähiger als etwa das Simpson-Verfahren.

2 Integration (Gauß)

Wenn wir auf die Eingabe der Schrittzahl verzichten und die Berechnungen beenden wollen, wenn sich für zwei aufeinander folgende n die Näherungen um höchstens 0,000001 unterscheiden, müssen wir das Programm ein wenig abändern:

```
10 INPUT "Integrationsgrenzen a,b";a,b
20 n=1:y=0:w=SQR(0.6)
30 n=n+1:h=(b-a)/2/n:g=0:d=w*h:z=a-h
40 FOR i=1 TO n:z=z+2*h
50 FOR j=-1 TO 1:x=z+j*d:GOSUB 120
60 g=g+(8-j*j*3)*f
70 NEXT j:NEXT i
80 g=g*h/9
90 IF ABS(y-g)>0.000001 THEN LET y=g:GOTO 30
100 PRINT "I=";g
110 END
120 f=
130 RETURN
```

Beispiel 4: (siehe Bsp. 1) $\int_0^{\pi/2} \sqrt{1+\cos^2 x}\, dx$

```
120 f=SQR(1+cos(x)*cos(x))
```

a,b = 0,1.570796327 ergibt I = 1.9100989

Beispiel 5: (siehe **BIN** 45 (1))

Bogenlänge der Kurve $\vec{x}(t) = (4t^3, 3t^4)$, $0 \le t \le 1$

Es ist $\dot{\vec{x}}(t) = (12t^2, 12t^3) = 12t^2 \cdot (1,t)$. Zu berechnen ist also:

$$\int_0^1 12t^2 \sqrt{1+t^2}\, dt$$

```
120 f=12*x*x*sqr(1+x*x)
```

a,b = 0,1 ergibt I = 5.04190068

3 Integration (Gauß mit Def. elementarer Funktionen)

Falls der Rechner die wichtigsten elementaren Funktionen (Kreis- und Hyperbelfunktionen samt ihren Umkehrungen, Logarithmus) nicht einprogrammiert hat, kann man das vorige Programm zu folgendem ausbauen. Die möglichen Definitionen elementarer Funktionen wird in **(3) elementare Funktionen** behandelt.

```
10 CLS: PRINT"    Bestimmtes Integral"
20 PRINT:PRINT"  elementarer Funktionen"
30 PRINT:PRINT"    (Gauss - Verfahren)"
40 PRINT:PRINT
50 INPUT "Integrand f(x) in Zeile 200 eingegeben? (j/n)";e$
60 PRINT:PRINT:IF e$="n" THEN LIST 200
70 REM **** Definition elem. Funkt.****
80 DEF FNcot(x)=1/TAN(x)
90 DEF FNarcsin(x)=
   -(x*x=1)*PI*SGN(x)/2-(x*x<1)*ATN(x/SQR(1-x*x-(x*x=1)))
100 DEF FNarccos(x)=PI/2-FNarcsin(x)
110 DEF FNarccot(x)=PI/2-ATN(x)
120 DEF FNcosh(x)=(EXP(x)+EXP(-x))/2
130 DEF FNsinh(x)=(EXP(x)-EXP(-x))/2
140 DEF FNtanh(x)=FNsinh(x)/FNcosh(x)
150 DEF FNarsinh(x)=LOG(x+SQR(1+x*x))
160 DEF FNarcosh(x)=LOG(x+SQR(x*x-1))
170 DEF FNartanh(x)=LOG((1+x)/(1-x))/2
180 DEF FNarcoth(x)=LOG((1+x)/(x-1))/2
190 REM ***** Eingabe: Integrand *****
200 DEF FNf(x)=
210 REM ***** Integration (Gauss) *****
220 INPUT "Integrationsgrenzen a,b";a,b
230 n=1:q=0:w=SQR(0.6)
240 n=n+1:g=0:s=(b-a)/2/n:d=w*s:x=a-s
250 FOR i=1 TO n:x=x+2*s
260 g=g+5*FNf(x-d)+8*FNf(x)+5*FNf(x+d)
270 NEXT i:g=g*s/9
280 IF ABS(q-g)>0.000001 THEN LET q=g:GOTO 240
290 PRINT:PRINT"I=";g:END
```

Beispiel 6: (siehe **REP** 176 (34))

$$F_1 = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cosh(x)\cos(x)dx$$

Aus Symmetriegründen (der Integrand ist eine gerade Funktion) ist $F_1 = 2\cdot\int_0^{\frac{\pi}{2}} \cosh(x)\cos(x)dx = 2\cdot I$

Wir programmieren:

```
200 DEF FN f(x) = FN cosh(x)*cos(x).
```

und starten das Programm:

```
a,b = 0,1.570796327        ergibt
```

I = 1.25458922 , das ist $\frac{1}{2}\cosh(\frac{\pi}{2})$, also $F_1 = \cosh\frac{\pi}{2}$.

Beispiel 7: Man berechne die vom Radiusvektor $r(\varphi) = \operatorname{arcosh}(\varphi+1)$, $0 \leq \varphi \leq 2\pi$ überstrichene Fläche.

Die von der in Polarkoordinaten gegebenen Kurve $r(\varphi) = \operatorname{arcosh}(\varphi+1)$ eingeschlossene Fläche ist (s. **BIN** 46 (3)).

$F = \frac{1}{2}\cdot\int_0^{2\pi} r^2(\varphi)d\varphi = \frac{1}{2}\cdot\int_0^{2\pi} \operatorname{arcosh}^2(x+1)dx.$

Wir programmieren:

```
200 DEF FN f(x) = 0.5*FNarcosh(x+1)*FNarcosh(x+1)
```

und starten das Programm:

```
a,b = 0 , 6.28318531          ergibt        I =13.0304054
```

Die Programmzeilen 80 bis 180 lassen sich natürlich auch bei anderen Programmen einbauen.

---

4 Gebietsintegral (Gauß)

Folgendes Programm berechnet Näherungen für Gebietsintegale. Es wird zuerst über y und dann über x integriert. Das heißt, die Grenzen für y sind Funktionen von x, während die Grenzen für x konstant sein müssen.

Die untere Grenze ya=g(x) von y ist in Zeile 230 und die obere Grenze yb=h(x) von y ist in Zeile 240 zu programmieren.

der Integrand f=f(x,y) ist in Zeile 210 zu programmieren.

Um Tipparbeit zu sparen, können die Zeilen 10 bis 70 weggelassen werden.

```
10 CLS
20 INPUT "untere Grenze ya in Zeile 230 und obere Grenze yb
          in Zeile 240 eingegeben (j/n)";e$

30 PRINT
40 IF ASC(e$)>106 THEN LIST 230-240:PRINT
50 INPUT "Integrand f(x,y) in Zeile 210 eingegeben (j/n)";f$
60 PRINT
70 IF ASC(f$)>106 THEN LIST 210:PRINT
80 INPUT "untere Grenze fuer x: a";a
90 INPUT "obere Grenze fuer x: b";b
100 n=1:q=0:w=SQR(0.6)
110 n=n+1:g=0
120 s=(b-a)/2/n:d=w*s:z=a-s
130 FOR i=1 TO n:z=z+2*s
140 FOR j=-1 TO 1:x=z+j*d:GOSUB 230
150 g=g+(8-j*j*3)*f
160 NEXT j:NEXT i
170 g=g*s/9
180 IF ABS(q-g)>0.00001 THEN LET q=g:GOTO 110
190 PRINT: PRINT "I=";g
200 END
210 f=
220 RETURN
230 ya=
240 yb=
```

```
250 m=1
260 m=m+1:gg=0
270 ss=(yb-ya)/2/m:dd=w*ss:zz=ya-ss
280 FOR ii=1 TO m:zz=zz+2*ss
290 FOR jj=-1 TO 1:y=zz+jj*dd:GOSUB 210
300 gg=gg+(8-jj*jj*3)*f
310 NEXT jj:NEXT ii
320 gg=gg*ss/9
330 IF ABS(qq-gg)>0.000001 THEN LET qq=gg:GOTO 260
340 f=gg
350 RETURN
```

Beispiel 8: $\int_{x=0}^{\pi}\int_{y=0}^{\sin x} dydx$

Wir programmieren:

```
230 ya = 0
240 yb = sin(x)
210 f = 1
```

Nun starten wir das Programm und beantworten die beiden Fragen mit j.

```
a = 0
b = 3.14159265        ergibt        I = 2.0000006
```

Natürlich ist $\int_{0}^{\pi}\int_{0}^{\sin x} dydx = \int_{0}^{\pi} \sin x dx = 2$

Beispiel 9: (siehe **REP** 277 (1))

$$\int_{x=2}^{3}\int_{y=x}^{x^2} (x+2y)dydx$$

```
230 ya = x
240 yb = x*x
210 f = x+2*y

a = 2
b = 3                 ergibt        I = 45.7833334    (= 2747/60 )
```

Beispiel 10: (siehe **REP** 279 (8)) Wir setzen R=1.

$$V^* = \int_{x=0}^{1} \int_{y=-\sqrt{\frac{1}{4}-(x-\frac{1}{2})^2}}^{\sqrt{\frac{1}{4}-(x-\frac{1}{2})^2}} \sqrt{1-(x^2+y^2)}\ dydx$$

Aus Symmetriegründen ist $\frac{1}{2} V^* = \int_{0}^{1} \int_{0}^{\sqrt{\frac{1}{4}-(x-\frac{1}{2})^2}} \sqrt{1-(x^2+y^2)}\ dydx$

Wir programmieren:

```
230 ya = 0
240 yb = SQR(0.25-(x-0.5)*(x-0.5))
210 f = SQR(1-x*x-y*y)

a = 0
b = 1                ergibt          I = 0.301437282
```

Das hat ganz schön gedauert und das Ergebnis ist nicht überwältigend.

Nach dem **REP** ist $\frac{1}{2}V^* = \frac{1}{2}\cdot\frac{1}{3}\cdot(\pi-\frac{4}{3}) = 0.301376.$

Man muß jedoch bedenken, daß während des Programmablaufs -da n bis 12 läuft - $3\sum_{n=2}^{12} n = 231$ einfache Integrationen nach Gauß auszuführen waren!

Versucht man es, wie bei der Lösung im **REP** vorgeschlagen, mit Polarkoordinaten, so hat man (wieder aus Symmetriegründen) folgendes Integral zu lösen (**REP** 250):

$$\int_{x=0}^{\frac{\pi}{2}} \int_{y=0}^{\cos x} \sqrt{1-y^2}\ y\ dydx$$

Hierbei sind nur $3\cdot(2+3)=15$ einfache Integrationen auszuführen; aber für x dicht bei 0 liegt cos(x) dicht bei 1 und $\int_{0}^{\cos x} \sqrt{1-y^2}\ y\ dy$ benötigt bis zu m=25 Iterationen, um mit der geforderten Genauigkeit berechnet zu werden.
Das geht nicht schneller.

PBZ (Partialbruchzerlegung)

echt gebrochener rationaler Funktionen

Folgendes Programm liefert die **PBZ** (**REP**, 54) rationaler Funktionen. Es beruht auf der konsequenten Anwendung (im Komplexen) der Zuhaltemethode (**REP**, 55). Es werden also keine Gleichungssysteme gelöst.

Voraussetzungen:

- Die rationale Funktion muß echt gebrochen sein, d.h. der Grad des Zählers muß echt kleiner sein als der Grad des Nenners.
- Das Nennerpolynom muß in Potenzen von Linearfaktoren und in Potenzen von quadratischen Faktoren zerlegt sein.

Dabei können folgende Programme hilfreich sein:

Rechnen mit Polynomen: Zerlegen einer gebrochenen rationalen Funktion in Polynom + echt gebrochener rationaler Funktion (Division mit Rest von Polynomen).

Polynomzerlegung: Berechnung der Nullstellen von Polynomen bis fünften Grades, Zerlegen eines Polynoms bis fünften Grades in im Reellen unzerlegbare - d.h. lineare und quadratische - Faktoren.

näherungsweise Nullstellenbestimmung: Bestimmung von Nullstellen des Nennerpolynoms.

Horner: Division eines Polynoms durch einen Linearfaktor, "Abspalten" eines Linearfaktors.

Hinweise:

Die quadratischen Faktoren im Nenner brauchen nicht unzerlegbar zu sein. Siehe Beispiele 2, 3, 4, 5.

Verschiedene Nennerfaktoren dürfen gemeinsame Nullstellen haben. Siehe Beispiele 2, 3, 5, 6.

Die gebrochene rationale Funktion braucht nicht gekürzt zu sein. Siehe Beispiele 3, 4.

```
10 CLS
20 REM*****          Eingabe          *****
30 REM*****************************************************
40 PRINT"     Partialbruchzerlegung (PBZ)":PRINT
50 PRINT"echt gebrochener rationaler Funktionen":PRINT
60 PRINT"=====================================":PRINT:PRINT
70 INPUT "Nennergrad";n:n1=n:a=0:m=n+1:PRINT
80 DIM r(m),i(m),d(m),p(m),q(m),ip(m),iq(m),in(m),rn(m)
90 DIM k(n),z(n),l(n),a(n),b(n),n(n,2),z1(n,n),e(n,n),f(n,n)
100 INPUT "Anzahl der Potenzen von Linearfaktoren im Nenner";mr
110 IF mr>n THEN 340
120 IF mr=0 THEN 210
130 FOR i=1 TO mr:PRINT
140 PRINT i;".-ter Linearfaktor ( ax+b)^k"
150 PRINT
160 INPUT "a=";n(i,1)
170 INPUT "b=";n(i,0):PRINT
180 INPUT "k=";k(i):a=a+k(i)
190 IF a>n THEN 340
200 NEXT i:PRINT
210 INPUT "Anzahl der Potenzen von quadratischen Faktoren";mc
220 IF a+2*mc>n THEN 340
230 m=mr+mc
240 IF mc=0 THEN 330
250 PRINT: FOR i=mr+1 TO m
260 PRINT i-mr;".-ter quadr Faktor (ax^2+bx+c)^k":PRINT
270 INPUT "a=";n(i,2)
280 INPUT "b=";n(i,1)
290 INPUT "c=";n(i,0):PRINT
300 INPUT "k=";k(i):a=a+2*k(i)
310 IF a>n THEN 340
320 NEXT i
330 IF a=n THEN 360
340 PRINT:PRINT"Eingabefehler, noch einmal"
350 PRINT:CLEAR:CLS:GOTO 70
360 PRINT:  INPUT "Zaehlergrad";z:PRINT
370 IF z>=n THEN PRINT"echt gebrochen?":GOTO 360
```

```
380 PRINT"Koeffizienten des Zaehlers"
390 PRINT: FOR i=0 TO z
400 PRINT "z(";z-i;")=";:INPUT z(z-i)
410 NEXT i
420 REM******************************
430 REM*** eventuelle Zerlegung quadratischer Faktoren ***
440 f=1:d=0
450 FOR i=1 TO m
460 IF n(i,2)=0 THEN 570
470 d(i)=n(i,1)*n(i,1)-4*n(i,2)*n(i,0)
480 IF d(i)<0 THEN 570
490 d(i)=SQR(d(i))
500 x=(n(i,1)-d(i))/2:GOSUB 2080:n(i,0)=x
510 x=n(i,0)+d(i):GOSUB 2080:n(m+1+d,0)=x
520 n(i,1)=n(i,2):n(m+1+d,1)=n(i,2):f=f*n(i,2)↑k(i)
530 n(i,2)=0:k(m+1+d)=k(i)
540 IF d(i)=0 THEN LET k(i)=2*k(i)
550 mc=mc-1:d=d+1
560 IF d(i)=0 THEN LET d=d-1
570 NEXT i:m=m+d:mr=m-mc
580 FOR i=0 TO z:z(i)=z(i)*f:NEXT i
590 REM**** Nullstellen des Nenners ****
600 FOR i=1 TO m
610 IF n(i,2)=0 THEN 660
620 a(i)=-n(i,1)/n(i,2)/2
630 b(i)=SQR(n(i,0)/n(i,2)-a(i)*a(i))
640 l(i)=n(i,2)
650 GOTO 680
660 a(i)=-n(i,0)/n(i,1)
670 l(i)=n(i,1)
680 NEXT i
690 REM*** gleiche Nenner-Faktoren zusammenfassen ***
700 FOR i=1 TO m-1
710 FOR j=1 TO m-i
720 IF NOT (a(i)=a(i+j) AND b(i)=b(i+j)) THEN 760
730 k(i+j)=k(i)+k(i+j):w(i)=l(i+j)/l(i)
740 FOR l=0 TO z:z(l)=z(l)*w(i)↑k(i):NEXT l
```

```
750 j=m-i:k(i)=0
760 NEXT j
770 NEXT i
780 REM**** Nenner ausmultiplizieren ****
790 r=0:p(0)=1
800 FOR i=1 TO m
810 IF k(i)=0 THEN 870
820 s=2:IF n(i,2)=0 THEN s=1
830 FOR j=1 TO k(i)
840 q(2)=n(i,2):q(1)=n(i,1):q(0)=n(i,0)
850 GOSUB 1950
860 NEXT j
870 NEXT i
880 FOR i=0 TO n:rn(i)=p(i):NEXT i
890 REM************************************
900 REM*******  PBZ - Koeffizienten *******
910 REM************************************
920 FOR i=1 TO m
930 IF k(i)=0 THEN 1690
940 x=a(i):y=b(i)
950 REM****** Zuhalten ******
960 g=n:h=k(i)
970 FOR l=0 TO g
980 r(l)=rn(l):i(l)=0
990 NEXT l
1000 GOSUB 1820
1010 REM***** in Nenner einsetzen *****
1020 n=n-k(i):g=n:h=1
1030 FOR l=0 TO g
1040 rn(l)=r(l+1)/l(i)^k(i)
1050 in(l)=i(l+1)/l(i)^k(i)
1060 r(l)=rn(l):i(l)=in(l)
1070 NEXT l
1080 GOSUB 1820:u=r(0):v=i(0)
1090 IF b(i)=0 THEN 1200
1100 REM***** konj kompl Fakt zusammenfassen *****
1110 FOR l=0 TO g
```

```
1120 r(l)=rn(l):i(l)=in(l)
1130 NEXT l
1140 h=k(i):y=-b(i):GOSUB 1820
1150 n=n-k(i)
1160 FOR l=0 TO n
1170 rn(l)=r(l+1)
1180 NEXT l
1190 REM***** in Zaehler einsetzen *****
1200 FOR j=k(i) TO 1 STEP -1
1210 FOR l=0 TO z:r(l)=z(l):i(l)=0:NEXT l
1220 g=z:h=1:y=b(i):GOSUB 1820
1230 REM***** komplexe Koeffizienten e, f *****
1240 e(i,j)=(r(0)*u+i(0)*v)/(u*u+v*v)
1250 f(i,j)=(-r(0)*v+i(0)*u)/(u*u+v*v)
1260 IF b(i)=0 THEN f(i,j)=e(i,j):e(i,j)=0:GOTO 1470
1270 REM*** konj PB auf Hauptnenner, reelle e,f ***
1280  FOR l=0 TO j:z1(j,l)=0:NEXT l
1290 r=0:p(0)=e(i,j):ip(0)=f(i,j)
1300 FOR c=1 TO j
1310 s=1:q(1)=1:q(0)=-a(i)
1320 iq(1)=0:iq(0)=b(i)
1330 GOSUB 1950
1340 NEXT c
1350 FOR l=0 TO j:z1(j,l)=z1(j,l)+2*p(l):NEXT l
1360 x=a(i):y=b(i):h=1:g=j
1370 FOR l=0 TO j
1380 r(l)=z1(j,l):i(l)=0
1390 NEXT l
1400 GOSUB 1820:w=r(0)
1410 y=-b(i):g=j-1
1420 FOR l=0 TO g
1430 r(l)=r(l+1):i(l)=i(l+1)
1440 NEXT l:GOSUB 1820
1450 e(i,j)=r(0):f(i,j)=w-x*r(0)
1460 REM**** linke Seite Hauptnenner, kuerzen ****
1470 r=1:s=n:p(1)=-e(i,j):p(0)=-f(i,j)
1480 ip(1)=0:ip(0)=0
```

```
1490 FOR l=0 TO n
1500 q(l)=rn(l):iq(l)=0
1510 NEXT l:GOSUB 1950
1520 IF z>n THEN 1540
1530 z=n+1:IF b(i)=0 THEN z=n
1540 g=z:h=1:y=b(i)
1550 FOR l=0 TO g
1560 r(l)=z(l)+p(l)
1570 NEXT l:GOSUB 1820
1580 IF b(i)=0 THEN 1630
1590 z(g)=0: g=g-1:y=-b(i)
1600 FOR l=0 TO g
1610 r(l)=r(l+1):i(l)=i(l+1)
1620 NEXT l:GOSUB 1820
1630 z(g)=0: z=g-1
1640 FOR l=0 TO z:z(l)=r(l+1)/l(i):NEXT l
1650 REM**** neue u,v fuer neues j ****
1660 IF b(i)=0 THEN 1680
1670 ua=v/b(i)/2:v=-u/b(i)/2:u=ua
1680 NEXT j
1690 NEXT i
1700 REM***** finish *****
1710 FOR i=1 TO m
1720 FOR j=1 TO k(i)
1730 x=f(i,j):GOSUB 2080:f(i,j)=x
1740 NEXT j:NEXT i:IF mc=0 THEN 1790
1750 FOR i=mr+1 TO m
1760 FOR j=1 TO k(i)
1770 x=e(i,j):GOSUB 2080:e(i,j)=x
1780 NEXT j:NEXT i
1790 GOTO 2140
1800 REM******************************
1810 REM*****  Horner  *****
1820 h1=1
1830 FOR l=1 TO h:r(g+1)=0:i(g+1)=0
1840 FOR k=0 TO g
1850 r(g-k)=r(g-k)+x*r(g-k+1)-y*i(g-k+1)
```

```
1860 i(g-k)=i(g-k)+x*i(g-k+1)+y*r(g-k+1)
1870 NEXT k
1880 IF h1=h THEN 1930
1890 h1=h1+1:g=g-1
1900 FOR k=0 TO g
1910 r(k)=r(k+1):i(k)=i(k+1)
1920 NEXT k:NEXT l
1930 RETURN
1940 REM*****  Multiplikation zweier  Polynome *****
1950 t=r+s:FOR l=0 TO t:xa=0:ya=0
1960 a=(l-r+ABS(l-r))/2
1970 b=(l+s-ABS(l-s))/2
1980 FOR k=a TO b
1990 xa=xa+p(l-k)*q(k)-ip(l-k)*iq(k)
2000 ya=ya+p(l-k)*iq(k)+ip(l-k)*q(k)
2010 NEXT k
2020 r(l)=xa:i(l)=ya
2030 NEXT l:r=t
2040 FOR l=0 TO r:p(l)=r(l):ip(l)=i(l):NEXT l
2050 FOR l=r+1 TO n1:p(l)=0:ip(l)=0:NEXT l
2060 RETURN
2070 REM***** "Runden" *****
2080 x=x*10↑3
2090 IF (x-INT(x))<10↑-3 THEN x=INT(x):GOTO 2110
2100 IF INT(x)+1-x<10↑-3 THEN x=INT(x)+1
2110 x=x/10↑3
2120 RETURN
2130 REM****************************
2140 REM*******  Ausgabe  ********
2150 CLS:PRINT:PRINT"Partialbrueche:"
2160 PRINT:PRINT
2170 FOR i=1 TO m
2180 n0$=STR$(n(i,0)):m0$=n0$
2190 n1$=STR$(n(i,1)):m1$=n1$
2200 n2$=STR$(n(i,2))
```

```
2210 IF n(i,0)>=0 THEN LET n0$="+"+n0$
2220 IF n(i,1)>=0 THEN LET n1$="+"+n1$
2230 FOR j=k(i) TO 1 STEP -1
2240 f$=STR$(f(i,j)):e$=STR$(e(i,j))
2250 IF f(i,j)>=0 THEN LET f$="+"+f$
2260 IF n(i,2)=0 THEN PRINT"(";m1$;"x ";
     n0$;" )↑";-j;"  *  ";f(i,j):GOTO 2290
2270 PRINT"(";n2$;"x↑2 ";n1$;"x ";n0$;" )↑";-j;
2280 PRINT"  *  (";e$;"x ";f$;" )"
2290 NEXT j:PRINT:NEXT i
2300 END
```

Beispiel 1:

$$\frac{2x^7+8x^6+13x^5+20x^4+17x^3+16x^2+7x+7}{(x-1)^2(x^2+x+1)^2(x^2+2x+2)}$$

```
Nennergrad?  8
Anzahl der Potenzen von Linearfaktoren im Nenner?   1
1 - ter Linearfaktor  (ax+b)↑k
a=1
b=-1
k=2
Anzahl der Potenzen von quadratischen Faktoren?  2
1 - ter quadratischer Faktor (ax↑2+bx+c)↑k
a=1
b=1
c=1
k=2
2 - ter quadratischer Faktor (ax↑2+bx+c)↑k
a=1
b=2
c=2
k=1
Zählergrad?  7
Koeffizienten des Zählers
z(7) = 2
z(6) = 8
z(5) = 13
z(4) = 20
z(3) = 17
z(2) = 16
z(1) = 7
z(0) = 7
```

Es werden nun die Partialbrüche in folgender Form ausgegeben:

Partialbrüche:

(1x-1)↑-2 * (2)

(1x-1)↑-1 * (1)

(1x↑2+1x+1)↑-2 * (1)

(1x↑2+1x+1)↑-1 * (0)

(1x↑2+2x+2)↑-1 * (1x+3)

Damit ergibt sich folgende Partialbruchzerlegung:

$$= \frac{2}{(x-1)^2} + \frac{1}{x-1} + \frac{1}{(x^2+x+1)^2} + \frac{x+3}{x^2+2x+2}$$

Beispiel 2: $\dfrac{144}{(x-1)(x+1)(x^2-1)^2}$

Die quadratischen Faktoren im Nenner brauchen nicht unzerlegbar zu sein: $x^2-1 = (x+1)(x-1)$

Verschiedene Nennerfaktoren dürfen gemeinsame Nullstellen haben.

Nennergrad? 6
Anzahl der Potenzen von Linearfaktoren im Nenner? 2
1 - ter Linearfaktor (ax+b)↑k
a=1, b=-1, k=1

2 - ter Linearfaktor (ax+b)↑k
a=1, b=1, k=1

Anzahl der Potenzen von quadratischen Faktoren? 1
a=1, b=0, c=-1, k=2

Zählergrad? 0
Koeffizienten des Zählers
z(0) = 144

Partialbrüche:

(1x-1)↑-3 * (18)
(1x-1)↑-2 * (-27)
(1x-1)↑-1 * (27)

(1x+1)↑-3 * (-18)
(1x+1)↑-2 * (-27)
(1x+1)↑-1 * (-27)

Damit ergibt sich folgende Partialbruchzerlegung:

$$= \frac{18}{(x-1)^3} - \frac{27}{(x-1)^2} + \frac{27}{x-1} - \frac{18}{(x+1)^3} - \frac{27}{(x+1)^2} - \frac{27}{x+1}$$

Gibt man die gleiche rationale Funktion entsprechend in einer der folgenden Formen ein, so erhält man das gleiche Ergebnis:

$\frac{144}{(x-1)(x+1)(x^2-1)^2}$	Anzahl der Potenzen von Lin-Faktoren Anzahl der Pot von quadratischen Fakt	= 2 = 1
$\frac{144}{(x-1)^3(x+1)^3}$	"	= 2
	"	= 0
$\frac{144}{(x^2-1)(x^2-1)(x^2-1)}$	"	= 0
	"	= 3
$\frac{144}{(x^2-1)^3}$	"	= 0
	"	= 1
$\frac{144}{(x-1)(x-1)(x-1)(x+1)(x+1)(x+1)}$	"	= 6
	"	= 0

Beispiel 3: $\dfrac{14x^2-4x-64}{(2x+4)(x^2-x-6)}$

Die rationale Funktion ist nicht gekürzt:
Es ist $14x^2-4x-64 = (x+2)(14x-32)$

Der quadratische Faktor läßt sich zerlegen:
$x^2-x-6 = (x+2)(x-3)$

Die beiden verschiedenen Nennerfaktoren $(2x+4)$ und (x^2-x-6) haben die gemeinsame Nullstelle -2 und damit den gemeinsamen Faktor $(x+2)$.

Wir erhalten die Partialbruchzerlegung:

$$\frac{14x^2-4x-64}{(2x+4)(x^2-x-6)} = \frac{1}{x-3} + \frac{6}{x+2}$$

Das gleiche Ergebnis erhalten wir nach Kürzen für:

$$\frac{7x-16}{(x+2)(x-3)} = \frac{6}{x+2} + \frac{1}{x-3}$$

Beispiel 4: $\dfrac{2x^3-2x^2-2x+2}{(x-1)^3(x+1)^2}$

$$= \frac{2x-2}{(x-1)^2(x+1)} = \frac{2}{(x-1)(x+1)} = \frac{2}{x^2-1}$$

In welcher Form wir die rationale Funktion auch eingeben, wir erhalten die

Partialbruchzerlegung: $= \dfrac{1}{x-1} - \dfrac{1}{x+1}$

Beispiel 5: (vergleiche Beispiel 2)

$$\frac{144}{(2x^2-2)^3} = \frac{18}{(2x-2)^3} - \frac{13,5}{(2x-2)^2} + \frac{6,75}{(2x-2)} - \frac{18}{(2x+2)^3} - \frac{13,5}{(2x+2)^2} - \frac{6,75}{(2x+2)}$$

$$\frac{144}{(2x^2-2)(x^2-1)^2} = \frac{9}{(x-1)^3} - \frac{13,5}{(x-1)^2} + \frac{13,5}{(x-1)} - \frac{9}{(x+1)^3} - \frac{13,5}{(x+1)^2} - \frac{13,5}{(x+1)}$$

$$\frac{144}{(2x^2-2)(3x^2-3)^2} = \frac{27}{(3x-3)^3} - \frac{13,5}{(3x-3)^2} + \frac{4,5}{(3x-3)} - \frac{27}{(3x+3)^3} - \frac{13,5}{(3x+3)^2} - \frac{4,5}{(3x+3)}$$

Beispiel 6:

Für die gleiche rationale Funktion können sich - je nach Eingabe - verschiedene PBZ-Darstellungen ergeben:

$$\frac{108}{(2x+2)^2(3x+3)^3(5x+5)^2} = \frac{3125}{(5x+5)^7}$$

$$= \frac{108}{(5x+5)^2(3x+3)^3(2x+2)^2} = \frac{5,12}{(2x+2)^7}$$

$$= \frac{108}{(5x+5)^2(2x+2)^2(3x+3)^3} = \frac{87,48}{(3x+3)^7}$$

$$= \frac{36}{(5x+5)^2(2x+2)^2(3x+3)^2(x+1)} = \frac{0,04}{(x+1)^7}$$

Daß es sich in allen Fällen um die gleiche rationale Funktion handelt, kann man leicht nachprüfen:

Es ist z.B.: $3125(\frac{2}{5})^7 = 5,12$

Beispiel 7: Weitere PBZ zum Probieren:

$$\frac{x^{14}+2x^{13}-4x^{12}-4x^{11}+12x^{10}-7x^9-82x^8-94x^7-36x^6+25x^5+9x^4-12x^3-5x^2+2x+1}{(x+1)^6x^5(x-1)^4}$$

$$= \frac{1}{(x+1)^6} - \frac{1}{(x+1)^3} + \frac{1}{x^5} - \frac{2}{x^3} + \frac{1}{x} - \frac{3}{(x-1)^4} + \frac{2}{(x-1)^3}$$

$$\frac{64x^6+632x^5+712x^4-1914x^3-1371x^2+1773x+1656}{(2x+3)^3(4x-5)^2(3x^2+3x+3)}$$

$$= \frac{3}{(2x+3)^3} - \frac{1}{(2x+3)} + \frac{1}{(4x-5)^2} + \frac{2x+3}{3x^2+3x+3}$$

$$\frac{6x^2-8x-6}{(x+1)^2(x^2-2x+5)} = \frac{1}{(x+1)^2} - \frac{2}{(x+1)} + \frac{2x-1}{x^2-2x+5}$$

$$\frac{2x^{11}+2x^{10}+5x^8+10x^6+10x^4+5x^2+1}{x^{10}(x^2+1)^5} = \frac{1}{x^{10}} + \frac{2x+1}{(x^2+1)^5}$$

$$\frac{5x^2-10x+5}{(x-1)^2(x+3)^3} = \frac{5}{(x+3)^3}$$ rationale Funktion nicht gekürzt!

$$\frac{2x^4+3x^3-2x^2+7x-9}{(x^2+1)^3} = \frac{4x-5}{(x^2+1)^3} + \frac{3x-6}{(x^2+1)^2} + \frac{2}{x^2+1}$$

$$\frac{-4x^6+19x^5-25x^4+11x^3-36x^2-12x-35}{x(x^2-4x+5)(x^2+1)^2} = -\frac{7}{x} + \frac{3x-5}{x^2-4x+5} + \frac{2x-3}{(x^2+1)^2} - \frac{4}{x^2+1}$$

$$\frac{6x-7}{(2x-4)^2} = \frac{5}{(2x-4)^2} + \frac{3}{2x-4}$$

$$\frac{66x^2-159x+129}{(2x-4)^2(3x+9)} = \frac{5}{(2x-4)^2} + \frac{3}{2x-4} + \frac{12}{3x+9}$$

$$\frac{-64x^4-24x^3+320x^2-6x-123}{(2x+3)^3(4x-5)^2} = \frac{3}{(2x+3)^3} - \frac{1}{(2x+3)} + \frac{1}{(4x-5)^2}$$

Lineare Differentialgleichungen

1. Lineare homogene Differentialgleichung mit konstanten Koeffizienten n-ter Ordnung und $n \leq 5$. (REP 250).

Die Lösungen der Differentialgleichung

$$y^{(n)} + a_{n-1}y^{(n-1)} + \ldots + a_1 y' + a_0 y = 0$$

ermittelt man nach Lösung der charakteristischen Gleichung

$$x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 = 0 \ .$$

Aus den Lösungen dieser algebraischen Gleichung ergeben sich dann leicht Basislösungen der Differentialgleichung.

Demgemäß stimmt das folgende Programm weitgehend mit dem Programm 'Polynomzerlegung' überein.

```
10 CLS:PRINT"Bestimmung der Loesungen einer linearen"
20 PRINT"homogenen DGL hoechstens 5.Ordnung":PRINT
30 a=-1:c=-1:DIM z(3,1)
40 GOSUB 60:DIM d(n-1):DIM e(n+1):GOSUB 110
50 GOSUB 600:PRINT:END
60 PRINT"Ordnung der DGL ";:INPUT"n=";n:n0=n
70 IF(n<1 OR n>5) THEN 60
80 DIM l(n),g(n),m(n):PRINT"Koeffizienten :"
90 PRINT:FOR k=n TO 0 STEP -1
100 PRINT"a("k")=";:INPUT l(k):NEXT k:PRINT:RETURN
110 ON n GOTO 120,130,190,200,360
120 a=a+1:d(a)=-l(0)/l(1):RETURN
130 GOSUB 390:IF r=1 THEN 170
140 c=c+1:z(c,0)=s:z(c,1)=-v
150 c=c+1:z(c,0)=s:z(c,1)=v
160 p0=l(1):q0=l(0):RETURN
170 a=a+1:d(a)=x1
180 a=a+1:d(a)=x2:RETURN
190 GOSUB 510:a=a+1:d(a)=x:GOSUB 370:GOTO 130
200 DIM t(4):FOR k=0 TO 4
210 t(k)=l(k)/l(4):NEXT k
220 l(0)=t(1)*t(1)-t(0)*(4*t(2)-t(3)*t(3))
230 l(1)=t(1)*t(3)-4*t(0):l(2)=t(2)
```

```
240 l(4)=0:l(3)=1:n=3
250 GOSUB 510:GOSUB 370:GOSUB 390
260 IF r=0 THEN y=x:GOTO 280
270 y=x1:IF x<x1 THEN y=x
280 l(0)=t(0):l(1)=y:l(2)=1:n=2
290 GOSUB 390:q2=x1:q1=x2:IF w=0 THEN 320
300 p1=(t(3)*q1-t(1))/2/v
310 p2=(t(1)-t(3)*q2)/2/v:GOTO 340
320 l(1)=-t(3):l(0)=t(2)+y:GOSUB 390
330 p1=x1:p2=x2
340 l(1)=p1:l(0)=q1:GOSUB 130
350 l(1)=p2:l(0)=q2:GOSUB 130:RETURN
360 GOSUB 510:a=a+1:d(a)=x:GOSUB 370:GOTO 200
370 GOSUB 490:FOR k=0 TO n-1
380 l(k)=e(k+1):NEXT k:n=n-1:RETURN
390 s=-l(1)/l(2)/2
400 w=l(1)*l(1)-4*l(0)*l(2)
410 IF ABS(w)<10↑(-4) THEN w=0
420 v=SQR(ABS(w))/l(2)/2:IF w<0 THEN r=0:RETURN
430 x1=s-v:x2=s+v:r=1:RETURN
440 h=0:ON n-1 GOTO 480,450,460,470
450 h=l(3):GOTO 480
460 h=l(3)+x*l(4):GOTO 480
470 h=l(3)+x*(l(4)+x*l(5))
480 f=l(0)+x*(l(1)+x*(l(2)+x*h)):RETURN
490 e(n+1)=0:FOR k=0 TO n
500 e(n-k)=l(n-k)+e(n-k+1)*x:NEXT k:f=e(0):RETURN
510 s=SGN(l(0)/l(n)):b=10:x=0
520 b=b/10:s=-s:IF ABS(b)<10↑(-7) THEN RETURN
530 x=x+b*s
540 GOSUB 440
550 IF f=0 THEN RETURN
560 IF l(n)*f*s<0 THEN GOTO 530
570 GOTO 520
580 '
590 'Ausgabe der Basisloesungen
600 PRINT"Basisloesungen sind :":PRINT
```

```
610 IF a=-1 THEN 870
620 '
630 'Erfassen unterschiedlicher reeller Nullstellen
640 'und deren Vielfachheit
650 FOR k=0 TO a:x=d(k):GOSUB 970:d(k)=x:NEXT k
660 g(1)=d(0):m(1)=1:k=1:IF a=0 THEN 740
670 FOR i=1 TO a:FOR j=1 TO k
680 IF g(j)=d(i) THEN m(j)=m(j)+1:GOTO 710
690 NEXT j
700 k=k+1:g(k)=d(i):m(k)=1
710 NEXT i
720 '
730 'Loesungsfunktionen zu reellen Nullstellen
740 FOR i=1 TO k:FOR j= 1 TO m(i)
750 IF j=1 AND g(i)=0 THEN c$="1":GOTO 810
760 a$="exp("+STR$(g(i))+"*x)"
770 IF j=1 THEN c$=a$:GOTO 810
780 c$="x↑"+STR$(j-1)
790 IF g(i)=0 THEN 810
800 c$=c$+"*"+a$
810 PRINT c$
820 NEXT j:NEXT i
830 IF c=-1 THEN RETURN
840 '
850 'Loesungsfunktionen zu komplexen Nullstellen
860 a$=""
870 FOR k=1 TO c STEP 2:x=z(k,0):GOSUB 970:r=x
880 x=z(k,1):GOSUB 970:y=x
890 c$="cos("+STR$(y)+"*x)":s$="sin("+STR$(y)+"*x)"
900 IF r=0 THEN b$="":GOTO 920
910 b$="exp("+STR$(r)+"*x)*"
920 PRINT a$+b$+c$
930 PRINT a$+b$+s$
940 IF z(3,0)=z(1,0) AND z(3,1)=z(1,1) THEN a$="x*"
950 NEXT k:RETURN
960 '
970 IF (x-INT(x)<10↑(-7)) THEN x=INT(x):RETURN
```

```
980 IF (INT(x)+1-x<10↑(-7)) THEN x=INT(x)+1
990 RETURN
```

Beispiel 1 : $y^{IV} + y''' = 0$

Basislösungen : $1 , x , x^2 , e^{-x}$.

Beispiel 2 : $y^{IV} - 2y''' + 2y'' - 2y' + y = 0$

Basislösungen : $e^x , xe^x , \cos x , \sin x$.

Beispiel 3 : $y^{V} + 4y^{IV} + 14y''' + 20y'' + 25y' = 0$

Basislösungen : $1 , e^{-x}\cos 2x , e^{-x}\sin 2x$

$xe^{-x}\cos 2x , xe^{-x}\sin 2x$.

2. Lineare inhomogene Differentialgleichung mit konstanten Koeffizienten n-ter Ordnung. (REP 253).

In den Anwendungen treten meist Störfunktionen der folgenden Form auf :

$r(x) = B(x) \cdot e^{ax} \cdot \cos bx$ oder $r(x) = B(x) \cdot e^{ax} \cdot \sin bx$.

wobei B(x) ein Polynom bezeichnet.

Um eine spezielle Lösung der DGL zu erhalten, macht man einen Lösungsansatz

$$y_s = x^k \cdot \left(\sum_{j=0}^{m} c_j x^j \right) e^{sx}$$

mit komplexen Koeffizienten $c_j = p_j + iq_j$ und $s = a + ib$.

Einsetzen dieser Funktion y_s in die DGL führt auf ein lineares Gleichungssystem für die Zahlen p_j und q_j.

Bei geeigneter Wahl von k (k-fache Resonanz) erweist sich dieses System als lösbar.In den Zeilen 280-390 wird die Lösung berechnet. Je nach dem ,ob c=0 oder c=1 eingegeben wurde, gibt das Programm den Real- bzw. den Imaginärteil von y_s als spezielle Lösung an.

Die nötigen Eingaben für c , m , $b_m,..,b_0$, a , b ergeben sich aus der Form der Störfunktion.

```
c=0 : Störfunktion enthält 'cos' oder keine trig.Funktion
c=1 : Störfunktion enthält 'sin'
m   : Polynomgrad  b(m),...,b(0) Koeffizienten von B
a,b : entnimmt man den Argumenten von exp bzw sin,cos
```

```
10 CLS
18 '
19 ' Ueberschrift
20 GOSUB 510
21 '
29 ' Koeffizienteneingabe
30 INPUT"n=";n:DIM rt(n+1,n+1),it(n+1,n+1)
40 DIM d(n),e(n)
50 FOR k=n TO 0 STEP -1:PRINT"a("k")=";
60 INPUT rt(0,n-k+1):NEXT k
61 '
69 ' Eingabe der Stoerfunktion
70 GOSUB 550
80 INPUT"c=";c
90 INPUT"m=";m:DIM b(m),p(m),q(m)
100 PRINT"Koeffizienten von B(x)"
110 FOR k=m TO 0 STEP -1:PRINT"b("k")=";
120 INPUT b(k):NEXT k
130 GOSUB 600
140 INPUT"a=";u:INPUT"b=";v
141 '
149 ' komplexes Hornerschema
150 FOR i=1 TO n+1:FOR k=0 TO n+1-i
160 s=rt(i,k):t=it(i,k)
170 x=u*s-v*t:y=u*t+v*s
180 rt(i,k+1)=rt(i-1,k+1)+x
190 it(i,k+1)=it(i-1,k+1)+y:NEXT k
200 d(i-1)=rt(i,n-i+2)
210 e(i-1)=it(i,n-i+2):NEXT i
211 '
219 ' Feststellen der Resonanz
220 FOR i=0 TO n:IF (d(i)=0 AND e(i)=0) THEN r=r+1:NEXT i
230 GOSUB 610
240 PRINT"k=";r:PRINT
241 '
249 ' Berechnung der p(i) und q(i)
250 z=d(r)*d(r)+e(r)*e(r)
```

```
260 x=d(r)/z:y=-e(r)/z
270 h=r+m:GOSUB 480:g=f:h=m:GOSUB 480
280 p(m)=f*x*b(m)/g:q(m)=f*y*b(m)/g
290 IF m=0 THEN 400
300 k=0
310 k=k+1:s=0:t=0:a=(r+k-n+ABS(r+k-n))/2
320 FOR i=a TO k-1
330 h=r+m-i:GOSUB 480:g=f:h=m-k:GOSUB 480
340 s=s+g*(d(r+k-i)*p(m-i)-e(r+k-i)*q(m-i))/f
350 t=t+g*(d(r+k-i)*q(m-i)+e(r+k-i)*p(m-i))/f:NEXT i
360 h=r+m-k:GOSUB 480:g=f:h=m-k:GOSUB 480
370 p(m-k)=f*((b(m-k)-s)*x+t*y)/g
380 q(m-k)=f*((b(m-k)-s)*y-t*x)/g
390 IF k<m THEN 310
400 IF c=0 THEN 430
410 FOR k=0 TO m:g=p(m-k):p(m-k)=q(m-k)
420 q(m-k)=-g:NEXT k
421 '
429 ' Ausgabe der Loesung
430 GOSUB 620
440 PRINT:FOR k=m TO 0 STEP -1:PRINT"p("k")="p(k):NEXT k
450 IF v=0 THEN PRINT"Q(x)=0":END
460 PRINT:FOR k=m TO 0 STEP -1:q(k)=-q(k)
461 '
462 PRINT"q("k")="q(k):NEXT k:PRINT:END
470 ' Fakultaet
480 f=1:IF h=0 THEN RETURN
490 FOR j=1 TO h:f=f*j:NEXT j:RETURN
491 '
500 ' Textausgaben
510 PRINT"Bestimmung einer speziellen Loesung"
520 PRINT"einer linearen inhomogenen Dgl mit"
530 PRINT"konstanten Koeffizienten":PRINT
540 PRINT"Ordnung der Dgl ";:RETURN
550 PRINT:PRINT"Die Stoerfunktion hat die Form":PRINT
560 PRINT"B(x)*exp(ax)*cos(bx)  (c=0)   oder"
570 PRINT"B(x)*exp(ax)*sin(bx)  (c=1)"
```

```
580 PRINT"(B(x) ist ein Polynom vom Grad m)":RETURN
590 PRINT"Welcher Fall liegt vor :";:RETURN
600 PRINT"Wie lauten die Zahlen a,b ?":RETURN
610 PRINT"k-fache Resonanz :";:RETURN
620 PRINT"Die Loesung hat die Form :":PRINT
630 PRINT"x↑k*exp(ax)*[P(x)cos(bx)+Q(x)sin(bx)]"
640 PRINT"(P,Q Polynome vom Grad hoechstens m)":RETURN
```

Beispiele zur Eingabe von c, m, b_j, a, b

Störfunktion	c	m		a	b
$x^2 + 2x$	0	2	$b_2=1\ b_1=2\ b_0=0$	0	0
$x \cdot e^{3x}$	0	1	$b_1=1\ b_0=0$	3	0
$x^2 \cdot \cos 2x$	0	2	$b_2=1\ b_1=0\ b_0=0$	0	2
$5x \cdot e^{2x} \cdot \sin 3x$	1	1	$b_1=5\ b_0=0$	2	3
$4e^{-x} \cdot \sin 2x$	1	0	$b_0=4$	-1	2

Beispiel 1 $\quad y''' - y'' + 4y' - 4y = 6 \cdot \sin x$

Eingabe : $a_3=1\ a_2=-1\ a_1=4\ a_0=-4\ c=1\ m=0\ b(0)=6\ a=0\ b=1$

Ausgabe : $k=0\ p(0)=-1\ q(0)=-1$

eine spezielle Lösung : $y_s = x\uparrow 0 \cdot e^{0x} \cdot (-1 \cos x + (-1) \sin x)$

Ergebnis : $y_s = -\cos x - \sin x$

Beispiel 2 $\quad y^{(V)} + 2y''' + y' = x \cdot \cos x$

Eingabe : $a_5=1\ a_4=0\ a_3=2\ a_2=0\ a_1=1\ a_0=0$
$c=0\ m=1\ b(1)=1\ b(0)=0\ a=0\ b=1$

Ausgabe : $k=2\ p(1)=0\ p(0)=-0.25\ q(1)=-4.16667E-02\ q(0)=0$

Nach Umwandlung der Dezimalzahlen in Brüche ergibt sich :

Lösung : $y_s = x^2 \cdot \left(-\frac{1}{4} \cos x - \frac{1}{24} x \cdot \sin x \right)$

Beispiel 3 $\quad y'' - 4y' + 5y = 5 \cdot e^x \cdot \cos x$

Eingabe : $a_2=1\ a_1=-4\ a_0=5\ c=0\ m=0\ b(0)=5\ a=1\ b=1$

Ausgabe : $k=0\ p(0)=1\ q(0)=-2$

Lösung : $y_s = e^x \cdot (\cos x - 2 \cdot \sin x)$

Beispiel 4 $y''' + 6y' + 20y = 2x \cdot e^x \cdot \sin 3x$

Eingabe : $a_3=1$ $a_2=0$ $a_1=6$ $a_0=20$ c=1 m=1 b(1)=2 b(0)=0 a=1 b=3

Ausgabe : p(1)=-2.77778E-02 p(0)=-9.25926E-03
k=1
q(1)=-2.77778E-02 q(0)=2.77778E-02

Nach Umwandlung in Brüche (Kehrwerte bilden) ergibt sich die

Lösung : $y_s = x \cdot e^x \cdot \left[\left(-\frac{3}{108}x + \frac{3}{108}\right) \sin 3x + \left(-\frac{3}{108}x - \frac{1}{108}\right) \cos 3x \right]$

Beispiel 5 $y^{IV} + y = x^2 + 1$

Eingabe : $a_4=1$ $a_3=a_2=a_1=0$ $a_0=1$ c=0 m=2 b(2)=1 b(1)=0 b(0)=1
a=b=0

Ausgabe : k=0 p(2)=1 p(1)=0 p(0)=1

Lösung : $y_s = x^2 + 1$.

Zusammenfassung :

Es gibt eine spezielle Lösung der Form

$$y_s = x^k \cdot e^{ax} \cdot (P(x)\cos bx + Q(x)\sin bx) ,$$

wobei P und Q Polynome vom Grad $\leq$ m sind.

Das Programm berechnet die Zahl k sowie die Koeffizienten von P und Q.

3. Lineare Differentialgleichung mit konstanten Koeffizienten n-ter Ordnung und n $\leq$ 5.

Dieses Programm ist eine Zusammenfassung der vorangehenden Programme 1 und 2.
Die Berechnung einer speziellen Lösung kann für unterschiedliche Störfunktionen wiederholt werden, ohne daß die Koeffizienten nochmals eingegeben werden müssen.

```
10 CLS:PRINT"Bestimmung der Loesungen einer linearen"
20 PRINT"DGL mit konstanten Koeffizienten"
30 PRINT"hoechstens 5. Ordnung":PRINT
40 a=-1:c=-1:DIM zc(3,1)
50 GOSUB 80:DIM az(n),xc(n-1),sd(n+1)
60 DIM rt(n+1,n+1),ik(n+1,n+1),d(n),e(n),yu(n),vb(n)
70 GOSUB 110:GOSUB 170:GOSUB 640:GOTO 960
```

```
80 PRINT"Ordnung der DGL ";:INPUT"n=";n:n0=n
90 IF n<1 OR n>5 THEN 80
100 RETURN
110 PRINT"Koeffizienten :"
120 FOR k=n TO 0 STEP -1
130 PRINT"a("k")=";:INPUT az(k)
140 rt(0,n-k+1)=az(k)
150 NEXT k
160 PRINT:RETURN
170 ON n GOTO 180,190,250,260,420
180 a=a+1:xc(a)=-az(0)/az(1):RETURN
190 GOSUB 450:IF r=1 THEN 230
200 c=c+1:zc(c,0)=s:zc(c,1)=-v
210 c=c+1:zc(c,0)=s:zc(c,1)=v
220 p0=az(1):q0=az(0):RETURN
230 a=a+1:xc(a)=x1
240 a=a+1:xc(a)=x2:RETURN
250 GOSUB 570:a=a+1:xc(a)=x:GOSUB 430:GOTO 190
260 DIM p(4):FOR k=0 TO 4
270 p(k)=az(k)/az(4):NEXT k
280 az(0)=p(1)*p(1)-p(0)*(4*p(2)-p(3)*p(3))
290 az(1)=p(1)*p(3)-4*p(0):az(2)=p(2)
300 az(4)=0:az(3)=1:n=3
310 GOSUB 570:GOSUB 430:GOSUB 450
320 IF r=0 THEN y=x:GOTO 340
330 y=x1:IF x<x1 THEN y=x
340 az(0)=p(0):az(1)=y:az(2)=1:n=2
350 GOSUB 450:q2=x1:q1=x2:IF w=0 THEN 380
360 p1=(p(3)*q1-p(1))/2/v
370 p2=(p(1)-p(3)*q2)/2/v:GOTO 400
380 az(1)=-p(3):az(0)=p(2)+y:GOSUB 450
390 p1=x1:p2=x2
400 az(1)=p1:az(0)=q1:GOSUB 190
410 az(1)=p2:az(0)=q2:GOSUB 190:RETURN
420 GOSUB 570:a=a+1:xc(a)=x:GOSUB 430:GOTO 260
430 GOSUB 550:FOR k=0 TO n-1
440 az(k)=sd(k+1):NEXT k:n=n-1:RETURN
```

```
450 s=-az(1)/az(2)/2
460 w=az(1)*az(1)-4*az(0)*az(2)
470 IF ABS(w)<10↑(-4) THEN w=0
480 v=SQR(ABS(w))/az(2)/2:IF w<0 THEN r=0:RETURN
490 x1=s-v:x2=s+v:r=1:RETURN
500 h=0:ON n-1 GOTO 540,510,520,530
510 h=az(3):GOTO 540
520 h=az(3)+x*az(4):GOTO 540
530 h=az(3)+x*(az(4)+x*az(5))
540 f=az(0)+x*(az(1)+x*(az(2)+x*h)):RETURN
550 sd(n+1)=0:FOR k=0 TO n
560 sd(n-k)=az(n-k)+sd(n-k+1)*x:NEXT k:f=sd(0):RETURN
570 s=SGN(az(0)/az(n)):b=10:x=0
580 b=b/10:s=-s:IF ABS(b)<10↑(-7) THEN RETURN
590 x=x+b*s
600 GOSUB 500
610 IF f=0 THEN RETURN
620 IF az(n)*f*s<0 THEN GOTO 590
630 GOTO 580
640 PRINT"Basisloesungen sind :"
650 IF a=-1 THEN 840
660 FOR k=0 TO a:x=xc(k):GOSUB 930:xc(k)=x:NEXT k
670 yu(1)=xc(0):vb(1)=1:k=1:IF a=0 THEN 730
680 FOR i=1 TO a:FOR j=1 TO k
690 IF yu(j)=xc(i) THEN vb(j)=vb(j)+1:GOTO 720
700 NEXT j
710 k=k+1:yu(k)=xc(i):vb(k)=1
720 NEXT i
730 FOR i=1 TO k:FOR j= 1 TO vb(i)
740 IF j=1 AND yu(i)=0 THEN c$="1":GOTO 800
750 a$="exp("+STR$(yu(i))+"*x)"
760 IF j=1 THEN c$=a$:GOTO 800
770 c$="x↑"+STR$(j-1)
780 IF yu(i)=0 THEN 800
790 c$=c$+"*"+a$
800 PRINT c$
810 NEXT j:NEXT i
```

```
820 IF c=-1 THEN RETURN
830 a$=""
840 FOR k=1 TO c STEP 2:x=zc(k,0):GOSUB 930:r=x
850 x=zc(k,1):GOSUB 930:y=x
860 c$="cos("+STR$(y)+"*x)":s$="sin("+STR$(y)+"*x)"
870 IF r=0 THEN b$="":GOTO 890
880 b$="exp("+STR$(r)+"*x)*"
890 PRINT a$+b$+c$
900 PRINT a$+b$+s$
910 IF zc(3,0)=zc(1,0) AND zc(3,1)=zc(1,1) THEN a$="x*"
920 NEXT:RETURN
930 x=1000*x:IF x-INT(x)<10↑(-4) THEN x=INT(x):GOTO 950
940 IF INT(x)+1-x<10↑(-4) THEN x=INT(x)+1
950 x=x/1000:RETURN
960 PRINT: PRINT"Abbruch mit ´e´"
970 PRINT"Berechnung einer speziellen Loesung :"
980 PRINT "Weiter mit Leertaste"
990 e$=INKEY$:IF e$="" THEN 990
1000 PRINT:IF e$="e" THEN END
1010 PRINT"Die Stoerfunktion hat die Form  :":PRINT
1020 PRINT"B(x)*exp(ax)*cos(bx)   (c=0)   oder"
1030 PRINT"B(x)*exp(ax)*sin(bx)   (c=1)":PRINT
1040 PRINT"(B(x) ist ein Polynom vom Grad m)":PRINT
1050 PRINT"Welcher Fall liegt vor :";
1060 INPUT"c=";c
1070 INPUT"Polynomgrad m=";m
1080 PRINT"Koeffizienten von B(x)"
1090 FOR k=m TO 0 STEP -1:PRINT"b("k")=";
1100 INPUT b(k):NEXT k
1110 PRINT"Wie lauten die Zahlen a,b ?"
1120 INPUT"a=";u:INPUT"b=";v
1130 n=n0
1140 FOR i=1 TO n+1:FOR k=0 TO n+1-i
1150 s=rt(i,k):t=ik(i,k)
1160 x=u*s-v*t:y=u*t+v*s
1170 rt(i,k+1)=rt(i-1,k+1)+x
1180 ik(i,k+1)=ik(i-1,k+1)+y:NEXT k
```

```
1190 d(i-1)=rt(i,n-i+2)
1200 e(i-1)=ik(i,n-i+2):NEXT i:r=0
1210 FOR i=0 TO n:IF d(i)=0 AND e(i)=0 THEN r=r+1:NEXT i
1220 PRINT"k-fache Resonanz : k=";r:PRINT
1230 z=d(r)*d(r)+e(r)*e(r)
1240 x=d(r)/z:y=-e(r)/z
1250 h=r+m:GOSUB 1490:g=f:h=m:GOSUB 1490
1260 p(m)=f*x*b(m)/g:q(m)=f*y*b(m)/g
1270 IF m=0 THEN 1380
1280 k=0
1290 k=k+1:s=0:t=0:a=(r+k-n+ABS(r+k-n))/2
1300 FOR i=a TO k-1
1310 h=r+m-i:GOSUB 1490:g=f:h=m-k:GOSUB 1490
1320 s=s+g*(d(r+k-i)*p(m-i)-e(r+k-i)*q(m-i))/f
1330 t=t+g*(d(r+k-i)*q(m-i)+e(r+k-i)*p(m-i))/f:NEXT i
1340 h=r+m-k:GOSUB 1490:g=f:h=m-k:GOSUB 1490
1350 p(m-k)=f*((b(m-k)-s)*x+t*y)/g
1360 q(m-k)=f*((b(m-k)-s)*y-t*x)/g
1370 IF k<m THEN 1290
1380 IF c=0 THEN 1410
1390 FOR k=0 TO m:g=p(m-k):p(m-k)=q(m-k)
1400 q(m-k)=-g:NEXT k
1410 PRINT"Die Loesung hat die Form :":PRINT
1420 PRINT"x↑k*exp(ax)*[P(x)cos(bx)+Q(x)sin(bx)]"
1430 PRINT"(P,Q Polynome vom Grad hoechstens m)":PRINT
1440 FOR k=m TO 0 STEP -1:PRINT"p("k")="p(k):NEXT k
1450 IF v=0 THEN PRINT"Q(x)=0":GOTO 960
1460 PRINT:FOR k=m TO 0 STEP -1:q(k)=-q(k)
1470 PRINT"q("k")="q(k):NEXT k
1480 PRINT:GOTO 960
1490 f=1:IF h=0 THEN RETURN
1500 FOR j=1 TO h:f=f*j:NEXT j:RETURN
```

Beispiel 1 $y''' - 5y'' + 9y' - 5y = 16 \cdot \sin x$

Eingabe : n=3 a(3)=1 a(2)=-5 a(1)=9 a(0)=-5

Ausgabe : Basislösungen e^x , $e^{2x} \cdot \cos x$, $e^{2x} \cdot \sin x$

Eingabe : c=1 m=0 b(0)=16 a=0 b=1

Ausgabe : k=0 p(0)=-2 q(0)=0

Ergebnis : $y = c_1 e^x + c_2 e^{2x} \cos x + c_3 e^{2x} \sin x - 2\cos x.$

Beispiel 2 $y''' - 3y' + 2y = 5 + 6e^x$

Eingabe : n=3 a(3)=1 a(2)=0 a(1)=-3 a(0)=2

Ausgabe : Basislösungen e^{-2x} , e^x , $x \cdot e^x$

a) Störfunktion $r(x) = 5$

Eingabe : c=0 m=0 b(0)=5 a=0 b=0

Ausgabe : k=0 p(0)=2.5 q(0)=0

Lösung : $y_s = 2.5$

b) Störfunktion $r(x) = 6e^x$

Eingabe : c=0 m=0 b(0)=6 a=1 b=0

Ausgabe : k=2 p(0)=1 q(0)=0

Lösung : $y_s = x^2 \cdot e^x$

Gesamtlösung :

$$y = c_1 e^{-2x} + c_2 e^x + c_3 x \cdot e^x + \frac{5}{2} + x^2 e^x.$$

Beispiel 3 $y^{IV} + y'' = x^2 + e^{2x}$

Basislösungen : 1 , x , cos x , sin x

a) Störfunktion $r(x) = x^2$

Lösung : $y_s = x^2(\frac{1}{12}x^2 - 1)$

b) Störfunktion $r(x) = e^{2x}$

Lösung : $y_s = \frac{1}{20}e^{2x}$

Gesamtlösung :

$$y = c_1 + c_2 x + c_3 \cos x + c_4 \sin x + x^2(\frac{1}{12}x^2 - 1) + \frac{1}{20}e^{2x}$$

Runge -Kutta Verfahren

Schon einfach aussehende Differentialgleichungen können zwar lösbar sein, jedoch nicht-elementare Lösungsfunktionen besitzen (z.B. $y'=x^2+y^2$).Man ist dann auf Näherungsverfahren angewiesen.
Das Runge-Kutta Verfahren behandelt das Anfangswertproblem

(A) $\mathbf{y}' = \mathbf{f}(x,\mathbf{y})$, $\mathbf{y}(x_a) = \mathbf{y_0}$ für $x \in [x_a, x_e]$

y steht dabei für m Funktionen y(1),..,y(m) und demgemäß **f** für Funktionen $f_1,..,f_m$ sowie $\mathbf{y_0}$ für ya(1),..,ya(m).

Die Funktionen $f_1,..,f_m$ sind ab Zeile 380 zu programmieren.(ys(k) steht dort für y(k)')

Beispiel 1 $y' = -2xy^2$, $y(-1)=0,5$ für $x \in [-1,3]$

Hier ist m=1 und somit y(1) = y , ys(1) = y'
Programmiert wird :

```
380 ys(1)=-2*x*y(1)*y(1)
```

Nach Programmstart ist einzugeben :

```
      m = 1
  xa,xe = -1,3  Intervallgrenzen
      n =  Die Schrittzahl
  ya(1) = 0.5  Anfangswert
```

Beispiel 2 $y' = e^{-x} \cdot z$ $y(0) = 0$
$z' = z - e^{x} \cdot y$ $z(0) = 1$ $x \in [0,2]$

Hier ist m=2 und somit y(1)=y, ys(1)=y'
y(2)=z, ys(2)=z'

Programmiert wird :

```
380 ys(1)=exp(-x)*y(2)
390 ys(2)=y(2)-exp(x)*y(1)
```

Nach Programmstart ist einzugeben :

```
      m = 2
  xa,xe = 0,2  Intervallgrenzen
      n = Die Schrittzahl
  ya(1) = 0
          Anfangswerte
  ya(2) = 1
```

Eine Anfangswertaufgabe mit einer Differentialgleichung höherer Ordnung wird auf den Typ (A) zurückgeführt.

Beispiel 3

$y'' = -2y^2 - 4xyy'$, $y(-1)=y'(-1)=0,5$ für $x \in [-1,1]$

Man setzt $y(1)=y$ und $y(2)=y'$

Dann ist : $ys(1)=y' =y(2)$ und

$ys(2)=y'' =-2y(1)*y(1) -4x*y(1)*y(2)$

Programmiert wird

```
380 ys(1)=y(2)
390 ys(2)=-2*y(1)*y(1)-4*x*y(1)*y(2)
```

Nach Programmstart ist einzugeben :

$m = 2$

$x_a, x_e = -1,1$

n = Die Schrittzahl

$ya(1) = 0.5$

$ya(2) = 0.5$

Das Intervall $[x_a, x_e]$ wird durch $x_k = x_a + \frac{x_e - x_a}{n} \cdot k$, $0 \leq k \leq n$, in Teilintervalle aufgeteilt und es werden Näherungswerte für die Funktionswerte $y_i(x_0), \ldots . y_i(x_n)$ berechnet.

```
10 CLS:PRINT"Runge - Kutta Verfahren fuer ein"
20 PRINT"DGL-System 1.Ordnung mit m Gleichungen":PRINT
30 INPUT "m=";m:PRINT
40 PRINT"Integrationsintervall: xa bis xe"
50 PRINT:INPUT "xa,xe=";xa,xe:PRINT
60 DIM y(m),ya(m),ys(m),r(4,m),t(4),f(m),fa(m),v(4)
70 INPUT "Anzahl der Teilintervalle n=";n:PRINT
80 PRINT"Anfangswerte ya(1) bis ya(";m;")":PRINT
90 FOR l=1 TO m
100 PRINT"ya("l")=";:INPUT ya(l):fa(l)=ya(l)
110 f(l)=0:NEXT l
120 h=(xe-xa)/n:t(1)=2:t(2)=2:t(3)=1:t(4)=1
130 v(0)=14:v(1)=64:v(2)=24:v(3)=64:PRINT
140 FOR k=0 TO n:PRINT:FOR l=1 TO m
150 PRINT"y"l"("xa")="ya(l);
```

```
160 y(l)=ya(l):NEXT l:x=xa
170 FOR i=1 TO 4:GOSUB 380
180 IF i=1 THEN GOSUB 280
190 x=xa+h/t(i):FOR l=1 TO m
200 r(i,l)=h*ys(l):y(l)=ya(l)+r(i,l)/t(i)
210 NEXT l:NEXT i
220 FOR l=1 TO m:ya(l)=ya(l)
230 ya(l)=ya(l)+(r(1,l)+2*r(2,l)+2*r(3,l)+r(4,l))/6
240 NEXT l:xa=xa+h:NEXT k
250 END
260 '
270 'Fehlerabschaetzung
280 kf=k MOD 4:FOR l=1 TO m
290 f(l)=f(l)+v(kf)*ys(l):NEXT l
300 IF kf>0 OR k=0 THEN 350 ELSE PRINT
310 FOR l=1 TO m
320 fa(l)=fa(l)+f(l)*h/45:f(l)=14*ys(l)
330 IF k>0 THEN PRINT"Fehler y"l"ca."ya(l)-fa(l);
340 NEXT l:IF k>0 THEN PRINT
350 RETURN
360 '
370 ' DGL-System
380 'ys(1)=
390 'ys(2)=
400 'ys(3)=
410 RETURN
```

Die sich auf die jeweils vorangehenden Werte beziehende Fehlerangabe ist mit Vorsicht zu verstehen : Bezeichnet L die Lipschitzkonstante von f, so entspricht der angegebene dem tatsächlichen Fehler in dem Fall, daß

$$(x_e - x_a)\ L < 3\ .$$

Rechnung zu Beispiel 1

$$y' = -2xy^2\ ,\ y(-1)=0,5 \quad \text{für } x \in [-1,3]$$

Diese Anfangswertaufgabe läßt sich leicht exakt lösen (REP 235). Lösung ist $y = 1/(1+x^2)$.

Um einen Vergleich zwischen Näherung und exaktem Wert zu ermöglichen, wird der PRINT-Befehl in Zeile 150 ergänzt

```
150 PRINT ......ya(1);tab(30) 1/(1+xa*xa);
```

Als Anzahl der Teilintervalle wird n=10 eingegeben.

	y Näherung	y exakt
y 1 (-1)	= 0.5	0.5
y 1 (-0.6)	= 0.734945238	0.735294117
y 1 (-0.2)	= 0.960802542	0.961538462
y 1 (0.2)	= 0.960802542	0.961538462
y 1 (0.6)	= 0.734374653	0.735294117
Fehler y 1 ca.	1.52414E-02	
y 1 (1)	= 0.499793986	0.5
y 1 (1.4)	= 0.337970321	0.337837838
y 1 (1.8)	= 0.236013903	0.235849057
y 1 (2.2)	= 0.171359459	0.171232877
Fehler y 1 ca.	1.53598E-02	
y 1 (2.6)	= 0.128953778	0.128865979
y 1 (3)	= 0.100059809	1.00000E-01

Im Verhältnis zu der doch relativ großen Schrittweite ist dies eine beachtlich gute Näherung.Durch Vergrößern von n läßt sich der maximale Fehler bis unter 10^{-8} herunterdrücken.
Dieses Beispiel 1 ist allerdings auch (numerisch) gutartig, im Gegensatz zum später vorgestellten Beispiel 4.

Rechnung zu Beispiel 2

$y'=e^{-x}\cdot z$, $z'=z-e^{x}\cdot y$, $y(0)=0$, $z(0)=1$, $x \in [0,2]$

Die exakte Lösung ist : $y = \sin x$, $z = e^{x}\cdot \cos x$.

Um einen Vergleich zwischen Näherung und exaktem Wert zu ermöglichen, werden die exakten Werte in einer zusätzlichen Tabelle ausgedruckt.

Als Anzahl der Teilintervalle wird n=10 eingegeben.

y Näherung	z Näherung
y 1 (0) = 0	y 2 (0) = 1
y 1 (0.2)= 0.198668287	y 2 (0.2)= 1.19705823
y 1 (0.4)= 0.389415823	y 2 (0.4)= 1.37406745
y 1 (0.6)= 0.564638197	y 2 (0.6)= 1.5038713
y 1 (0.8)= 0.717349953	y 2 (0.8)= 1.5505699
Fehler y 1 ca.-9.38750E-06	Fehler y 2 ca. 9.18424E-06
y 1 (1) = 0.841463074	y 2 (1) = 1.46872738
y 1 (1.2)= 0.932029684	y 2 (1.2)= 1.20312152
y 1 (1.4)= 0.985439305	y 2 (1.4)= 0.689326464
y 1 (1.6)= 0.999562785	y 2 (1.6)=-0.144518675
Fehler y 1 ca.-2.64167E-05	Fehler y 2 ca. 2.22244E-05
y 1 (1.8)= 0.973837179	y 2 (1.8)=-1.37434489
y 1 (2) = 0.909288187	y 2 (2) =-3.07473572

x	y exakt	z exakt
0	0	1
0.2	0.198669331	1.19705602
0.4	0.389418342	1.37406154
0.6	0.564642474	1.50385954
0.8	0.717356091	1.5505493
1	0.841470985	1.46869394
1.2	0.932039086	1.20307011
1.4	0.98544973	0.689250751
1.6	0.999573603	-0.144626181
1.8	0.973847631	-1.37449258
2	0.909297427	-3.07493232

Rechnung zu Beispiel 3

$y'' = -2y^2 - 4xyy'$, $y(-1)=y'(-1)=0,5$ für $x \in [-1,1]$

Die exakte Lösung ist y=1/(1+x*x).

Wenn nur die Lösungsfunktion (und nicht ihre Ableitungen) interessiert, wird man den Ausdruck der Ableitungswerte verhindern :

```
140 .....:FOR l=1 TO m:IF l>1 THEN 160
325 IF l>1 THEN 340
```

Für einen Vergleich zwischen Näherung und exaktem Wert wird Zeile 150 ergänzt zu

```
150 .......ya(l);tab(30) 1/(1+xa*xa);
```

Als Anzahl der Teilintervalle wird n=10 eingegeben.

y Näherung	y exakt
y 1 (-1) = 0.5	0.5
y 1 (-0.8)= 0.609757688	0.609756098
y 1 (-0.6)= 0.735301374	0.735294117
y 1 (-0.4)= 0.862069741	0.862068965
y 1 (-0.2)= 0.961493119	0.961538462
Fehler y 1 ca.-1.82809E-04	
y 1 (-1.16415E-10) = 0.999914124	1
y 1 (0.2)= 0.961506923	0.961538462
y 1 (0.4)= 0.862120171	0.862068965
y 1 (0.6)= 0.735375844	0.735294118
Fehler y 1 ca.-4.70144E-04	
y 1 (0.8)= 0.609844915	0.609756098
y 1 (1) = 0.500100829	0.5

Anmerkung : Der Rechner arbeitet intern nicht mit 0,2 sondern mit einer Approximation durch eine Dualzahl.Daraus erklärt sich (Zeile 7 der Tabelle)

$$-1 + 0{,}2 + 0{,}2 + 0{,}2 + 0{,}2 + 0{,}2 = -1{,}16415E\text{-}10 .$$

Beispiel 4 $y' = 10y - 11e^{-x}$, $y(0)=1$

Die exakte Lösung läßt sich leicht berechnen (REP 238) : $y = e^{-x}$.

Programmiert wird :

```
150 ......ya(l);tab(30) exp(-xa);
380 ys(1)=10*y(1)-11*exp(-x)
```

Eingegeben wird : m=1 xa,xe = 0,1 n=10 ya(1)=1

y Näherung	y exakt
y 1 (0) = 1	1
y 1 (0.1)= 0.90473141	0.904837418
y 1 (0.2)= 0.818347727	0.818730753
y 1 (0.3)= 0.739694065	0.740818221
y 1 (0.4)= 0.667196923	0.670320046
Fehler y 1 ca.-1.97635E-04	
y 1 (0.5)= 0.598001141	0.60653066
y 1 (0.6)= 0.525646559	0.548811636
y 1 (0.7)= 0.433788376	0.496585304
y 1 (0.8)= 0.279201308	0.449328964
Fehler y 1 ca. 4.94840E-04	
y 1 (0.9)=-5.42404E-02	0.40656966
y 1 (1) =-0.880190837	0.367879441

Man sieht, daß der Fehler zum rechten Rand des Intervalls stark zunimmt (und die Fehlerangabe deutlich vom tatsächlichen Fehler abweicht). Durch Vergrößern von n läßt sich auf dem Intervall $[0,1]$ eine annehmbare Näherung erreichen. Nimmt man jedoch das größere Intervall $[0,5]$,so nutzt auch die Wahl einer größeren Schrittzahl nichts : Die Abweichung zwischen Näherung und exakter Lösung wird ab etwa x=2 rasch sehr groß.

Das Verfahren gestattet also nicht, die (völlig harmlose) Lösung $y=e^{-x}$ in größeren Intervallen anzunähern.

Die Lösungsgesamtheit der vorgelegten Differentialgleichung ist nämlich $y=e^{-x} + c \cdot e^{10x}$, $c \in \mathbb{R}$. Die Anfangsbedingung 'filtert' aus dieser Schar die Funktion $y=e^{-x}$ heraus. Durch unvermeidbare Rechenungenauigkeiten kommt man von der exakten Lösung ab und der Einfluß von $c \cdot e^{10x}$ führt zum Auseinanderlaufen von Hoffnung und Wirklichkeit.

Wer näher interessiert ist, lese z.B. den Abschnitt 4.2 in HEINHOLD,Einführung in die höhere Mathematik Teil 3.

Lineare Gleichungssysteme (LGS) $A \cdot \vec{x} = \vec{b}$

Ein LGS kann unlösbar sein, genau eine Lösung oder aber eine ein- oder mehrparametrige Lösungsschar besitzen.
Die Lösungsgesamtheit eines homogenen Gleichungssystems bildet einen r-dimensionalen Vektorraum, wobei r die Differenz zwischen der Variablenzahl und dem Rang der Koeffizientenmatrix ist.
Im Fall r=0 ist das homogene LGS eindeutig durch $\vec{0}$ lösbar, im Fall r>0 besitzt es eine r-parametrige Lösungsschar.
Die Lösungsgesamtheit eines inhomogenen LGS läßt sich beschreiben durch $\vec{x} = \vec{x}_s + \vec{x}_h$, wobei
$\vec{x}_s$ eine spezielle Lösung des inhomogenen Systems und
$\vec{x}_h$ eine beliebige Lösung des zugehörigen homogenen Systems ist.

1 Gaußsches Eliminationsverfahren

```
10 CLS:PRINT"Lineares Gleichungssystem":PRINT
20 INPUT "Zahl der Variablen  ";n
30 INPUT "Zahl der Gleichungen";m:PRINT
40 DIM a(m,n+1),c(n+1),z(n,n)
50 g=10↑(-6)
60 '
70 ' Koeffizienteneingabe
80 PRINT"Koeffizienteneingabe zeilenweise"
90 FOR i=1 TO m:FOR k=1 TO n
100 PRINT"a("i","k")";:INPUT a(i,k)
110 NEXT k
120 PRINT"b("i")";:INPUT a(i,n+1):PRINT
130 NEXT i
140 ' Eingabekontrolle
150 CLS:PRINT"Koeffizientenschema":PRINT
160 GOSUB 700
170 PRINT"Eingabe korrekt (j/n)"
180 e$=INKEY$:IF e$="" THEN 180
190 PRINT:IF e$="n" THEN CLS:GOTO 90
```

```
200 '
210 ' Eliminationsschritte
220 '
230 FOR k=1 TO n
240 FOR i=1 TO m
250 IF (a(i,0)>0 OR a(i,k)=0) THEN NEXT i:GOTO 360
260 a(0,k)=i:a(i,0)=k:f=f+1
270 FOR l=1 TO n+1:c(l)=a(i,l):NEXT l
280 FOR j=1 TO m
290 y=a(j,k)/c(k)
300 FOR l=1 TO n+1
310 a(j,l)=a(j,l)-c(l)*y
320 IF ABS(a(j,l))<g THEN a(j,l)=0
330 NEXT l:NEXT j
340 FOR l=1 TO n+1:a(i,l)=c(l)/c(k):NEXT l
350 GOSUB 700
360 NEXT k
370 ' Berechnung der Loesung
380 r=n-f
390 FOR k=1 TO n
400 i=a(0,k)
410 IF i<1 THEN v=v+1:z(k,v)=1:GOTO 470
420 z(k,0)=a(i,n+1)
430 FOR l=1 TO n
440 IF a(0,l)>0 THEN 460
450 w=w+1:z(k,w)=-a(i,l)
460 NEXT l:w=0
470 NEXT k
480 '
490 ' Test auf Loesbarkeit
500 FOR i=1 TO m
510 IF (a(i,0)=0 AND a(i,n+1)<>0) THEN PRINT"* unloesbar *":END
520 NEXT i
530 '
540 'Ausdruck der Loesung
550 PRINT"(";:FOR k=1 TO n
560 x=z(k,0):GOSUB 770:IF k<n THEN PRINT x"/";:NEXT k
```

```
570 PRINT x")"
580 IF r=0 THEN  PRINT"ist einzige Loesung":GOTO 670
590 PRINT"ist eine spezielle Loesung"
600 PRINT:PRINT"Der Loesungsraum des zugehoerigen"
610 PRINT"homogenen Systems hat die Dimension"r
620 PRINT"Eine Basis ist:"
630 FOR i=1 TO r:PRINT"(";
640 FOR k=1 TO n
650 x=z(k,i):GOSUB 770:IF k<n THEN PRINT x"/";:NEXT k
660 PRINT x")":NEXT i
670 END
680 '
690 ' Ausdruck der Koeffizientenmatrix
700 FOR p=1 TO m:FOR q=1 TO n+1
710 PRINT a(p,q);
720 IF q=n THEN PRINT" ";
730 NEXT q:PRINT
740 NEXT p:PRINT:RETURN
750 '
760 'Runden
770 x=1000*x:IF x-INT(x)<10↑(-4) THEN x=INT(x):GOTO 790
780 IF INT(x)+1-x<10↑(-4) THEN x=INT(x)+1
790 x=x/1000:RETURN
```

Das Programm endet mit einer der folgenden drei Möglichkeiten :

1) Ausgabe * unlösbar * oder
2) Ausgabe des eindeutig bestimmten Lösungsvektors oder
3) Ausgabe einer speziellen Lösung und einer Basis des Lösungsraums des zugehörigen homogenen Systems.

Beispiel 1	2x + 3y = 4
	4x + 6y = 4
Ergebnis :	* unlösbar *
Beispiel 2	-x + 8y + 3z = 2
	2x + 4y - z = 1
	-2x + y + 2z = -1
Ergebnis :	(5/-1/5) ist einzige Lösung

Beispiel 3

$$\begin{aligned} x - 3y + 5z - 2w &= -1 \\ -2x + 6y - 10z + 4w &= 2 \\ 3x - y + 3z - 10w &= -19 \\ x - y + 2z - 3w &= -5 \end{aligned}$$

Ergebnis : (-7/-2/0/0) ist eine spezielle Lösung.
Eine Lösungsbasis des homogenen Systems ist
(-0.5/1.5/1/0) (3.5/0.5/0/1)

Mit den Parametern s,t kann man die Lösung beschreiben durch

$\vec{x}$ = (-7/-2/0/0) + s(-0.5/1.5/1/0) + t(3.5/0.5/0/1).

Es folgt eine verkürzte Version des vorangehenden Programms

2 Kurzprogramm

```
10 INPUT "n=";n
20 INPUT "m=";m:f=0:w=0:v=0
30 DIM a(m,n+1),c(n+1)
40 FOR i=1 TO m:FOR k=1 TO n
50 PRINT"a(";i;",";k;")":INPUT a(i,k)
60 NEXT k
70 PRINT"b(";i;")":INPUT a(i,n+1)
80 NEXT i
90 '
100 FOR k=1 TO n
110 FOR i=1 TO m
120 IF ABS(a(i,k))<10↑(-5) THEN LET a(i,k)=0
130 IF (a(i,0)>0 OR a(i,k)=0) THEN NEXT i:GOTO 220
140 a(0,k)=i:a(i,0)=k:f=f+1
150 FOR l=1 TO n+1:c(l)=a(i,l):NEXT l
160 FOR j=1 TO m
170 y=a(j,k)/c(k)
180 FOR l=1 TO n+1
190 a(j,l)=a(j,l)-c(l)*y
200 NEXT l:NEXT j
210 FOR l=1 TO n+1:a(i,l)=c(l)/c(k):NEXT l
220 NEXT k
230 r=n-f
240 DIM z(n,r)
250 FOR k=1 TO n
260 i=a(0,k)
```

```
270 IF i<1 THEN LET v=v+1:z(k,v)=1:GOTO 330
280 z(k,0)=a(i,n+1)
290 FOR l=k TO n
300 IF a(0,l)>0 THEN 320
310 w=w+1:z(k,w)=-a(i,l)
320 NEXT l:w=0
330 NEXT k
340 FOR i=1 TO m
350 IF (a(i,0)=0 AND a(i,n+1)<>0) THEN PRINT"unloesbar":END
360 NEXT i
370 PRINT"(":FOR k=1 TO n
380 IF k<n THEN PRINT z(k,0);"/":NEXT k
390 PRINT z(k,0);")"
400 IF r=0 THEN  PRINT"eindeutig":END
410 PRINT"ist spez. Lsg"
420 PRINT"Dim ";r
430 PRINT"Basis:"
440 FOR i=1 TO r:PRINT"("
450 FOR k=1 TO n
460 IF k<n THEN PRINT z(k,i);"/":NEXT k
470 PRINT z(k,i);")":NEXT i
480 END
```

3 Kurzprogramm bei eindeutiger Lösbarkeit

(Sowie für den Fall der Unlösbarkeit)

```
10 INPUT "n=";n:f=0
20 DIM a(n,n+1),c(n+1),z(n)
30 FOR i=1 TO n:FOR k=1 TO n
40 PRINT"a(";i;",";k;")":INPUT a(i,k)
50 NEXT k
60 PRINT"b(";i;")":INPUT a(i,n+1)
70 NEXT i
80 FOR k=1 TO n
90 FOR i=1 TO n
100 IF ABS(a(i,k))<10↑(-5) THEN LET a(i,k)=0
110 IF (a(i,0)>0 OR a(i,k)=0) THEN 200
120 a(0,k)=i:a(i,0)=k:f=f+1
```

```
130 FOR l=1 TO n+1:c(l)=a(i,l):NEXT l
140 FOR j=1 TO n
150 y=a(j,k)/c(k)
160 FOR l=1 TO n+1
170 a(j,l)=a(j,l)-c(l)*y
180 NEXT l:NEXT j
190 FOR l=1 TO n+1:a(i,l)=c(l)/c(k):NEXT l
200 NEXT i
210 NEXT k
220 IF f<n THEN PRINT"nicht eind.":GOTO 290
230 FOR k=1 TO n
240 z(k)=a(a(0,k),n+1)
250 NEXT k
260 PRINT"(":FOR k=1 TO n
270 IF k<n THEN PRINT z(k);"/":NEXT k
280 PRINT z(k);")"
290 END
```

4 Pivot-Verfahren

Es gibt Gleichungssysteme, die (trotz korrekt programmiertem Lösungsverfahren) auf dem Rechner zu absurden Ergebnissen führen.

Ein Beispiel ist die Hilbertmatrix

$$a(i,k) = 1/(i+k) \quad \text{für } 1 \leq i,k \leq n$$

Definiert man $b(i) = \sum_{k=0}^{n} a(i,k)$ für $1 \leq i \leq n$

so hat das Gleichungssystem

$$\sum_{k=0}^{n} a(i,k)x_k = b(i) \quad \text{für } 1 \leq i \leq n$$

offenbar die Lösung (1,1,...,1).

Ab n=7 wird das mit dem hier angegebenen Programm ermittelte Ergebnis unbrauchbar.

Durch besondere Auswahl des 'Eliminationselements' kann man das Verfahren etwas stabilisieren.

1) Jede Zeile $(a_{i1},\ldots,a_{in})$ wird durch Division durch die Zahl $s(i) = \text{sqr}(a_{i1}^2 + .. + a_{in}^2)$ normiert.
2) Unter den für die Elimination noch in Betracht kommenden Elementen wählt man das maximale aus.

Das Pivotverfahren läßt sich leicht in das erste Programm dieses Abschnitts einbauen : Dazu sind die Zeilen 220-360 zu ersetzen durch :

```
220 '
225 FOR kk=1 TO n
230 ' Pivotsuche
235 FOR i=1 TO m
240 IF a(i,0)>0 THEN 255
245 s=0:FOR l=1 TO n
250 s=s+ABS(a(i,l))↑2:NEXT l:s(i)=SQR(s)
255 NEXT i
260 p=0:FOR k=1 TO n:FOR i=1 TO m
265 IF a(0,k)>0 THEN 295
270 IF a(i,k)=0 THEN 290
275 s=ABS(a(i,k))/s(i)
280 IF s<=p OR a(i,0)>0 OR s(i)=0 THEN 290
285 p=s:ip=i:kp=k
290 NEXT i
295 NEXT k
300 IF p=0 THEN 365
305 '
310 i=ip:k=kp
315 a(0,k)=i:a(i,0)=k:f=f+1
320 FOR l=1 TO n+1:c(l)=a(i,l):NEXT l
325 FOR j=1 TO m
330 y=a(j,k)/c(k)
335 FOR l=1 TO n+1
340 a(j,l)=a(j,l)-c(l)*y
345 IF ABS(a(j,l))<g THEN a(j,l)=0
350 NEXT l:NEXT j
355 FOR l=1 TO n+1:a(i,l)=c(l)/c(k):NEXT l
360 GOSUB 700:NEXT kk
```

5 LGS mit komplexen Koeffizienten

Die Eingabe von z.B. '3-2i' muß in der Form 3,-2 erfolgen, die Ausgabe erfolgt als (3,-2).

```
10 CLS:PRINT"Lineares Gleichungssystem mit komplexen Koeffizienten"
20 INPUT "Zahl der Variablen  ";n
30 INPUT "Zahl der Gleichungen";m:PRINT
40 DIM a1(m,n+1):DIM a2(m,n+1):DIM c1(n+1):DIM c2(n+1)
50 DIM z1(n,n):DIM z2(n,n)
60 g=10^(-7)
70 '
80 ' Koeffizienteneingabe
90 PRINT"Koeffizienteneingabe zeilenweise":PRINT
100 FOR i=1 TO m:FOR k=1 TO n
110 PRINT"a("i","k") :";:INPUT " Re,Im";a1(i,k),a2(i,k)
120 NEXT k
130 PRINT"b("i") :";:INPUT " Re,Im";a1(i,n+1),a2(i,n+1):PRINT
140 NEXT i
150 ' Eingabekontrolle
160 CLS:PRINT"Koeffizientenschema"
170 PRINT"Ausgabeform (Re z/Im z)":PRINT
180 GOSUB 890
190 PRINT"Eingabe korrekt (j/n)"
200 e$=INKEY$:IF e$="" THEN 200
210 PRINT:IF e$="n" THEN 100
220 '
230 'Eliminationsschritte
240 '
250 FOR k=1 TO n
260 FOR i=1 TO m
270 IF (a(i,0)>0 OR (a1(i,k)=0 AND a2(i,k)=0)) THEN NEXT i:GOTO 440
280 a(0,k)=i:a(i,0)=k:f=f+1
290 FOR l=1 TO n+1:c1(l)=a1(i,l):c2(l)=a2(i,l):NEXT l
300 FOR j=1 TO m:ne=c1(k)*c1(k)+c2(k)*c2(k)
310 y1=(a1(j,k)*c1(k)+a2(j,k)*c2(k))/ne
320 y2=(a2(j,k)*c1(k)-a1(j,k)*c2(k))/ne
```

```
330 FOR l=1 TO n+1
340 a1(j,l)=a1(j,l)-(c1(l)*y1-c2(l)*y2)
350 IF ABS(a1(j,l))<g THEN a1(j,l)=0
360 a2(j,l)=a2(j,l)-(c1(l)*y2+c2(l)*y1)
370 IF ABS(a2(j,l))<g THEN a2(j,l)=0
380 NEXT l:NEXT j
390 FOR l=1 TO n+1
400 a1(i,l)=(c1(l)*c1(k)+c2(l)*c2(k))/ne
410 a2(i,l)=(c2(l)*c1(k)-c1(l)*c2(k))/ne
420 NEXT l
430 GOSUB 890
440 NEXT k
450 '
460 ' Berechnung der Loesung
470 r=n-f
480 FOR k=1 TO n
490 i=a(0,k)
500 IF i<1 THEN v=v+1:z1(k,v)=1:GOTO 560
510 z1(k,0)=a1(i,n+1):z2(k,0)=a2(i,n+1)
520 FOR l=1 TO n
530 IF a(0,l)>0 THEN 550
540 w=w+1:z1(k,w)=-a1(i,l):z2(k,w)=-a2(i,l)
550 NEXT l:w=0
560 NEXT k
570 '
580 ' Test auf Loesbarkeit
590 FOR i=1 TO m
600 IF (a(i,0)<>0 OR (a1(i,n+1)=0 AND a2(i,n+1)=0)) THEN 620
610 PRINT"* unloesbar *":GOTO 810
620 NEXT i
630 '
640 ' Ausgabe der Loesung
650 PRINT"[";:FOR k=1 TO n
660 x=z1(k,0):GOSUB 840:u=x
670 x=z2(k,0):GOSUB 840:v=x
680 IF k<n THEN PRINT "("u"/"v")""/";:NEXT k
690 PRINT "("u"/"v")]"
700 IF r=0 THEN  PRINT"ist einzige Loesung":END
```

```
710 PRINT"ist eine spezielle Loesung"
720 PRINT:PRINT"Der Loesungsraum des zugehoerigen"
730 PRINT"homogenen Systems hat die Dimension"r
740 PRINT"Eine Basis ist:"
750 FOR i=1 TO r:PRINT"[";
760 FOR k=1 TO n
770 x=z1(k,i):GOSUB 840:u=x
780 x=z2(k,i):GOSUB 840:v=x
790 IF k<n THEN PRINT "("u"/"v")""/";:NEXT k
800 PRINT "("u"/"v")]":NEXT i
810 END
820 '
830 ' Runden
840 x=1000*x:IF x-INT(x)<10↑(-4) THEN x=INT(x):GOTO 860
850 IF INT(x)+1-x<10↑(-4) THEN x=INT(x)+1
860 x=x/1000:RETURN
870 '
880 ' Ausdruck der Koeffizienten
890 FOR p=1 TO m:FOR q=1 TO n+1
900 x=a1(p,q):GOSUB 840:u=x
910 x=a2(p,q):GOSUB 840:v=x
920 PRINT "("u"/"v")";
930 IF q=n THEN PRINT" ";
940 NEXT q:PRINT
950 NEXT p:PRINT
960 RETURN
```

Beispiel 1

$$\begin{aligned} (1+i)x + (1+2i)y + iz &= -1+4i \\ ix + y + 2z &= 2+2i \\ (1+2i)x + (2+i)y + 2iz &= -2+4i \end{aligned}$$

Lösung : $x=i \quad y=1 \quad z=1+i$

Beispiel 2

$$\begin{aligned} (1+i)x + (1+2i)y + iz &= -1+4i \\ ix + y + 2z &= 2+2i \\ (-2+i)x - (2-2i)y + (-1+2i)z &= -6+i \end{aligned}$$

Lösung : (Nach Umwandlung der Dezimalzahlen in Brüche)

$(\frac{1}{3} - \frac{2}{3}i \;/\; \frac{4}{3} + \frac{5}{3}i \;/\; 0) + t(\frac{2}{3} + i \;/\; -1 - \frac{2}{3}i \;/\; 1)$, wobei

t eine beliebige komplexe Zahl bedeutet.

Matrizengleichung A·X = B

Die Matrizengleichung A·X = B wird mit dem Gaußschen Eliminationsverfahren gelöst.

```
10 CLS:PRINT "Matrizengleichung A*X=B":PRINT
20 INPUT "Spaltenzahl von A";n
30 INPUT "Zeilenzahl von A";m
40 INPUT "Spaltenzahl von B";r
50 PRINT:PRINT"Die Zeilenzahl von B muss"m"sein"
60 DIM a(m,n+r+1):PRINT
70 g=10↑(-7)
80 '
90 ' Koeffizienteneingabe
100 PRINT "Koeffizienteneingabe von A und B"
110 PRINT"gemeinsam zeilenweise":PRINT
120 FOR i=1 TO m:FOR k=1 TO n
130 PRINT "A("i","k")";:INPUT a(i,k):NEXT k
140 FOR k=1 TO r:PRINT "B("i","k")";
150 INPUT a(i,n+k):NEXT k:PRINT
160 NEXT i
170 ' Eingabekontrolle
180 CLS:PRINT"Koeffizientenschema":PRINT
190 GOSUB 840
200 PRINT"Eingabe korrekt (j/n)"
210 e$=INKEY$:IF e$="" THEN 210
220 PRINT:IF e$="n" THEN CLS:GOTO 120
230 '
240 ' Eliminationsschritte
250 '
260 FOR k=1 TO n
270 FOR i=1 TO m
280 IF (a(i,0)>0 OR a(i,k)=0) THEN NEXT i:GOTO 390
290 a(0,k)=i:a(i,0)=k:f=f+1
300 FOR l=1 TO n+r:c(l)=a(i,l):NEXT l
310 FOR j=1 TO m
320 y=a(j,k)/c(k)
330 FOR l=1 TO n+r
340 a(j,l)=a(j,l)-c(l)*y
```

```
350 IF ABS(a(j,l))<g THEN a(j,l)=0
360 NEXT l:NEXT j
370 FOR l=1 TO n+r:a(i,l)=c(l)/c(k):NEXT l
380 GOSUB 840
390 NEXT k
400 '
410 'Berechnung der Loesung
420 r0=n-f
430 FOR k=1 TO n
440 i=a(0,k)
450 IF i<1 THEN v=v+1:y(k,v)=1:GOTO 510
460 FOR p=1 TO r:s(k,p)=a(i,n+p):NEXT p
470 FOR l=1 TO n
480 IF a(0,l)>0 THEN 500
490 w=w+1:y(k,w)=-a(i,l)
500 NEXT l:w=0
510 NEXT k
520 '
530 ' Test auf Loesbarkeit
540 FOR i=1 TO m
550 IF a(i,0)>0 THEN 590
560 FOR p=1 TO r
570 IF a(i,n+r)<>0 THEN PRINT"* unloesbar *":END
580 NEXT p
590 NEXT i
600 GOSUB 840
610 '
620 ' Ausdruck der Loesung
630 FOR k=1 TO n:FOR p=1 TO r
640 x=s(k,p):PRINT x;:NEXT p:PRINT
650 NEXT k
660 IF r0=0 THEN  PRINT"ist einzige Loesung":END
670 PRINT"ist eine spezielle Loesung"
680 PRINT:PRINT"Der Loesungsraum des zugehoerigen"
690 PRINT"homogenen Systems hat die Dimension"r0
700 PRINT"Eine Basis ist:"
710 FOR i=1 TO r0
```

```
720 FOR k=1 TO n
730 FOR p=1 TO r:PRINT y(k,i);:NEXT p:PRINT
740 NEXT k:PRINT
750 NEXT i
760 END
770 '
780 ' Runden
790 x=1000*x:IF x-INT(x)<10↑(-4) THEN x=INT(x):GOTO 810
800 IF INT(x)+1-x<10↑(-4) THEN x=INT(x)+1
810 x=x/1000:RETURN
820 '
830 ' Ausdruck der Koeffizienten
840 FOR p=1 TO m:FOR q=1 TO n+r
850 x=a(p,q):GOSUB 790:PRINT x;
860 IF q=n THEN PRINT" ";
870 NEXT q:PRINT:NEXT p:PRINT
880 RETURN
```

Beispiel 1

$$\begin{pmatrix} 1 & 2 \\ 2 & 3 \\ -1 & 1 \end{pmatrix} \cdot X = \begin{pmatrix} 4 & 2 & 3 \\ 6 & 4 & 5 \\ 2 & -2 & 0 \end{pmatrix}$$

$$X = \begin{pmatrix} 0 & 2 & 1 \\ 2 & 0 & 1 \end{pmatrix}$$ ist einzige Lösung.

Beispiel 2

$$\begin{pmatrix} 2 & -4 \\ -1 & 2 \\ 3 & -6 \end{pmatrix} \cdot X = \begin{pmatrix} 2 & -2 & 6 \\ -1 & 1 & -3 \\ 3 & -3 & 9 \end{pmatrix}$$

Es gibt eine 1-parametrige Lösungsschar :

$$X = \begin{pmatrix} 1 & -1 & 3 \\ 0 & 0 & 0 \end{pmatrix} + t \begin{pmatrix} 2 & 2 & 2 \\ 1 & 1 & 1 \end{pmatrix}$$, wobei t beliebig ist.

Produkte und Potenzen von Matrizen

1. Produkt zweier Matrizen mit reellen Elementen

```
10 CLS:PRINT"Matrizenprodukt":PRINT
20 INPUT "Zeilenzahl von A";m
30 INPUT "Spaltenzahl von A";n
40 PRINT"Die Zeilenzahl von B muss"n"sein"
50 INPUT "Spaltenzahl von B";p:PRINT
60 d1=m:IF n>m THEN d1=n
70 d2=n:IF p>n THEN d2=p
80 DIM a(m,n),b(d1,d2)
90 '
100 ' Hauptprogramm
110 r=m:s=n:k$="a":GOSUB 170
120 FOR i=1 TO m:FOR k=1 TO n:a(i,k)=b(i,k):NEXT k:NEXT i
130 r=n:s=p:k$="b":GOSUB 170
140 GOTO 280
150 '
160 ' Koeffizienteneingabe
170 FOR i=1 TO r:FOR k=1 TO s
180 PRINT k$+"("i","k")=";:INPUT b(i,k)
190 NEXT k:PRINT:NEXT i
200 ' Eingabekontrolle
210 FOR i=1 TO r:FOR k=1 TO s
220 PRINT b(i,k);:NEXT k:PRINT:NEXT i
230 PRINT:PRINT"Eingabe korrekt (j/n)"
240 e$=INKEY$:IF e$="" THEN 240
250 PRINT:IF e$="n" THEN 170
260 RETURN
270 '
280 ' Matrizenprodukt
290 PRINT"Ergebnis :":PRINT
300 FOR i=1 TO m:FOR k=1 TO p:s=0
310 FOR l=0 TO n:s=s+a(i,l)*b(l,k):NEXT l
320 PRINT s;:NEXT k:PRINT:NEXT i:PRINT
330 END
```

Beispiel :

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 1 & 0 \\ 3 & -1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 4 & -1 & 3 \\ 2 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 5 & 4 & 6 \\ 18 & -3 & 12 \\ 8 & -2 & 11 \end{pmatrix}$$

2. Potenzen einer quadratischen,reellen Matrix

```
10 CLS:PRINT"Matrixpotenz A↑p":PRINT
20 INPUT "Zeilenzahl von A";n
30 DIM a(n,n),b(n,n),c(n,n)
40 PRINT:GOSUB 180
50 INPUT "Exponent p";p:PRINT
60 IF p<1 THEN 50
70 '
80 ' Hauptprogramm
90 FOR h=1 TO p-1:GOSUB 300:NEXT h
100 PRINT"Ergebnis :":PRINT
130 FOR i=1 TO n:FOR k=1 TO n
140 PRINT c(i,k);:b(i,k)=a(i,k):c(i,k)=a(i,k)
150 NEXT k:PRINT:NEXT i
155 PRINT:GOTO 50
160 '
170 ' Koeffizienteneingabe
180 FOR i=1 TO n:FOR k=1 TO n
190 PRINT"a("i","k")=";:INPUT a(i,k)
200 b(i,k)=a(i,k):c(i,k)=a(i,k)
210 NEXT k:PRINT:NEXT i
220 ' Eingabekontrolle
230 FOR i=1 TO n:FOR k=1 TO n
240 PRINT a(i,k);:NEXT k:PRINT:NEXT i
250 PRINT:PRINT "Eingabe korrekt (j/n)"
260 e$=INKEY$:IF e$="" THEN 260
270 PRINT:IF e$="n" THEN 180
280 RETURN
290 '
300 ' Produkt
```

```
310 FOR i=1 TO n:FOR k=1 TO n:s=0
320 FOR l=0 TO n:s=s+a(i,l)*b(l,k):NEXT l
330 c(i,k)=s:NEXT k:NEXT i
340 ' Umbenennung
350 FOR i=1 TO n:FOR k=1 TO n
360 b(i,k)=c(i,k):NEXT k:NEXT i
370 RETURN
```

Beispiel 1

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 2 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad A^2 = \begin{pmatrix} 1 & 1 & 2 \\ 4 & 1 & 2 \\ 2 & 2 & 1 \end{pmatrix} \quad A^3 = \begin{pmatrix} 3 & 3 & 3 \\ 6 & 3 & 6 \\ 6 & 3 & 3 \end{pmatrix}$$

Beispiel 2

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A^2 = \begin{pmatrix} 1 & 4 & 10 & 20 \\ 0 & 1 & 4 & 10 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A^3 = \begin{pmatrix} 1 & 6 & 21 & 56 \\ 0 & 1 & 6 & 21 \\ 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A^4 = \begin{pmatrix} 1 & 8 & 36 & 120 \\ 0 & 1 & 8 & 36 \\ 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Invertierung von Matrizen

Eine quadratische Matrix ist genau dann invertierbar, wenn ihre Determinante ungleich null ist.
Zu einer Matrix der Zeilenzahl n bestimmt dieses Programm die inverse Matrix durch gleichzeitiges Lösen eines Gleichungssystems für n rechte Seiten. Die Determinante D(A) wird ebenfalls berechnet.
Da bei einer Matrix mit ganzzahligen Elementen die Matrix $D(A)*A^{-1}$ ebenfalls ganzzahlige Elemente besitzt, wird diese als Ergebnis ausgegeben.(REP 116).

1 Invertierung

```
10 CLS:PRINT "Matrixinvertierung":PRINT
20 INPUT "Zeilenzahl :";n
30 DIM a(n,2*n),c(2*n):d=1:s=1
40 g=10↑(-7)
50 '
60 ' Koeffizienteneingabe
70 PRINT:PRINT "Koeffizienteneingabe zeilenweise":PRINT
80 FOR i=1 TO n:FOR k=1 TO n
90 PRINT "a("i","k")";:INPUT a(i,k):NEXT k
100 a(i,n+i)=1:PRINT
110 NEXT i
120 ' Eingabekontrolle
130 CLS:PRINT"Koeffizientenschema":PRINT
140 m=n:GOSUB 470:m=2*n
150 PRINT"Eingabe korrekt (j/n)"
160 e$=INKEY$:IF e$="" THEN 160
170 PRINT:IF e$="n" THEN 80
180 '
190 ' Eliminationsschritte
200 FOR k=1 TO n
210 FOR i=1 TO n
220 IF ABS(a(i,k))<g THEN a(i,k)=0
230 IF (a(i,0)>0 OR a(i,k)=0) THEN NEXT i:GOTO 340
240 a(0,k)=i:a(i,0)=k:f=f+1
250 FOR l=1 TO 2*n:c(l)=a(i,l):NEXT l
260 d=d*c(k)
270 FOR j=1 TO n
```

```
280 y=a(j,k)/c(k)
290 FOR l=1 TO 2*n
300 a(j,l)=a(j,l)-c(l)*y
310 NEXT l:NEXT j
320 FOR l= 1 TO 2*n:a(i,l)=c(l)/c(k):NEXT l
330 GOSUB 470
340 NEXT k
360 '
365 IF f<n THEN PRINT"* nicht invertierbar *":GOTO 440
370 x=d:GOSUB 540:GOSUB 590:d=x*s
380 PRINT"D(A) ="d:PRINT
390 PRINT"Ausgegeben wird die Matrix "d"*A↑-1 :":PRINT
400 '
410 FOR k=1 TO n:FOR i=1 TO n
420 x=d*a(a(0,k),n+i):GOSUB 540:PRINT x;:NEXT i:PRINT
430 NEXT k
440 END
450 '
460 ' Koeffizientenausdruck
470 FOR p=1 TO n:FOR q=1 TO m
480 x=a(p,q):GOSUB 540:PRINT x;
490 IF q=n THEN PRINT" ";
500 NEXT q:PRINT:NEXT p:PRINT
510 RETURN
520 '
530 ' Runden
540 x=x*1000:IF x-INT(x)<10↑(-4) THEN x=INT(x):GOTO 560
550 IF INT(x)+1-x<10↑(-4) THEN x=INT(x)+1
560 x=x/1000:RETURN
570 '
580 ' Fehlstaende
590 FOR p=1 TO n-1:FOR q= p+1 TO n
600 IF a(0,p)<a(0,q) THEN 620
610 s=-s
620 NEXT q:NEXT p
630 RETURN
```

Durch Löschen der Zeilen 330,450-510 läßt sich der Ausdruck der einzelnen Rechenschritte vermeiden.

Beispiel :

$$\begin{pmatrix} 4 & -2 & 7 \\ 3 & 2 & 1 \\ 1 & 2 & 3 \end{pmatrix}^{-1} = \frac{1}{60}\begin{pmatrix} 4 & 20 & -16 \\ -8 & 5 & 17 \\ 4 & -10 & 14 \end{pmatrix}$$

Es folgt eine verkürzte Version des voranstehenden Programms. Die Elemente der Matrix $D(A)*A^{-1}$ werden zeilenweise nacheinander ausgegeben.

2 Kurzprogramm

```
10 INPUT "n=";n
20 DIM a(n,2*n),c(2*n):d=1:s=1
30 FOR i=1 TO n:FOR k=1 TO n
40 PRINT "a(";i;",";k;")":INPUT a(i,k):NEXT k
50 a(i,n+i)=1
60 NEXT i
70 FOR k=1 TO n
80 FOR i= 1 TO n
90 IF ABS(a(i,k))<10↑(-5) THEN LET a(i,k)=0
100 IF (a(i,0)>0 OR a(i,k)=0) THEN 200
110 a(0,k)=i:a(i,0)=k:f=f+1
120 FOR l=1 TO 2*n:c(l)=a(i,l):NEXT l
130 d=d*c(k)
140 FOR j=1 TO n
150 y=a(j,k)/c(k)
160 FOR l=1 TO 2*n
170 a(j,l)=a(j,l)-c(l)*y
180 NEXT l:NEXT j
190 FOR l=1 TO 2*n:a(i,l)=c(l)/c(k):NEXT l
200 NEXT i
210 NEXT k
220 IF f<n THEN PRINT"nicht inv.":GOTO 290
230 GOSUB 300:d=d*s
240 PRINT"D(A)=";d
250 PRINT d;"*A↑-1 :"
260 FOR k=1 TO n:FOR i=1 TO n
270 PRINT d*a(a(0,k),n+i):NEXT i
280 NEXT k
290 END
300 FOR p=1 TO n-1:FOR q= p+1 TO n
310 IF a(0,p)<a(0,q) THEN 330
320 s=-s
330 NEXT q:NEXT p
340 RETURN
```

Determinanten

1 Berechnung der Determinante

In diesem Programm wird zunächst die Matrix durch Zeilenumformung auf eine Form gebracht, die bis auf Zeilenvertauschung eine Diagonalform ist. Die Determinante ist dann, bis aufs Vorzeichen, das Produkt dieser 'Diagonalelemente'. Das Vorzeichen ergibt sich aus den nötigen Zeilenvertauschungen und wird in den Zeilen 500-540 bestimmt.

```
10 CLS:PRINT "Determinantenberechnung":PRINT
20 INPUT "Zeilenzahl :";n
30 DIM a(n,n):DIM c(n):d=1:s=1
40 '
50 ' Koeffizienteneingabe
60 PRINT:PRINT "Koeffizienteneingabe zeilenweise":PRINT
70 FOR i=1 TO n:FOR k=1 TO n
80 PRINT "a("i","k")";:INPUT a(i,k):NEXT k:PRINT
90 NEXT i
100 ' Eingabekontrolle
110 CLS:PRINT"Koeffizientenschema":PRINT
120 GOSUB 390:PRINT
130 PRINT"Eingabe korrekt (j/n)"
140 e$=INKEY$:IF e$="" THEN 140
150 PRINT:IF e$="n" THEN 70
160 '
170 ' Eliminationsschritte
180 FOR k=1 TO n
190 FOR i=1 TO n
200 IF ABS(a(i,k))<10↑(-5) THEN a(i,k)=0
210 IF (a(i,0)>0 OR a(i,k)=0) THEN 300
220 a(0,k)=i:a(i,0)=k:f=f+1
230 FOR l=1 TO n:c(l)=a(i,l):NEXT l
240 d=d*c(k)
250 FOR j=i TO n
260 y=a(j,k)/c(k)
270 FOR l=1 TO n
280 a(j,l)=a(j,l)-c(l)*y
290 NEXT l:NEXT j
300 NEXT i
310 NEXT k
320 IF f<n THEN x=0:GOTO 350
330 '
340 x=d:GOSUB 450:GOSUB 500:x=x*s
350 PRINT"Die Determinante von A ist "x
360 END
370 '
380 ' Ausdruck Koeffizientenschema
390 FOR p=1 TO n:FOR q=1 TO n
400 PRINT a(p,q);:NEXT q:PRINT
410 NEXT p
420 RETURN
```

```
430 '
440 ' Runden
450 x=x*1000:IF x-INT(x)<10↑(-4) THEN x=INT(x):GOTO 470
460 IF INT(x)+1-x<10↑(-4) THEN x=INT(x)+1
470 x=x/1000:RETURN
480 '
490 ' Fehlstaende
500 FOR p=1 TO n-1:FOR q=p+1 TO n
510 IF a(0,p)<a(0,q) THEN 530
520 s=-s
530 NEXT q:NEXT p
540 RETURN
```

Beispiel :

Berechnen Sie

$$\begin{vmatrix} 3 & 2 & 2 & 3 \\ 1 & -1 & 1 & -2 \\ 3 & 3 & 3 & -4 \\ 3 & 2 & 2 & 8 \end{vmatrix}$$

Ergebnis : Die Determinante ist -30.

2 Kurzprogramm

```
10 INPUT "n=";n
20 DIM a(n,n),c(n):d=1:s=1
30 FOR i=1 TO n:FOR k=1 TO n
40 PRINT "a(";i;",";k;")":INPUT a(i,k):NEXT k
50 NEXT i
60 FOR k=1 TO n
70 FOR i= 1 TO n
80 IF ABS(a(i,k))<10↑(-5) THEN LET a(i,k)=0
90 IF (a(i,0)>0 OR a(i,k)=0) THEN 180
100 a(0,k)=i:a(i,0)=k:f=f+1
110 FOR l=1 TO n:c(l)=a(i,l):NEXT l
120 d=d*c(k)
130 FOR j=i TO n
140 y=a(j,k)/c(k)
150 FOR l=1 TO n
160 a(j,l)=a(j,l)-c(l)*y
170 NEXT l:NEXT j
180 NEXT i
190 NEXT k
200 IF f<n THEN LET x=0:GOTO 220
210 GOSUB 240:x=d*s
220 PRINT"Det(A)=";x
230 END
240 FOR p=1 TO n-1:FOR q=p+1 TO n
250 IF a(0,p)<a(0,q) THEN 270
260 s=-s
270 NEXT q:NEXT p
280 RETURN
```

Eigenwerte und Eigenvektoren

1. Eigenwertproblem für 2 x 2 Matrizen

Diese Programm berechnet für eine 2 x 2 Matrix mit reellen Koeffizienten :

- die charakteristische Gleichung
- die Eigenwerte
- die zu reellen Eigenwerten gehörigen Eigenvektoren.

```
10 CLS
20 PRINT"Eigenwertaufgabe fuer 2*2-Matrizen":PRINT
30 DIM p(3),z(2)
40 DIM a(2,2),b(2,2),y(2,2)
50 '
60 ' Hauptprogramm
70 GOSUB 150:GOSUB 280:GOSUB 380
80 GOSUB 320:GOSUB 430
90 x=z(1):GOSUB 530
100 IF w=0 THEN 120
110 x=z(2):GOSUB 530
120 END
130 '
140 ' Koeffizienteneingabe
150 FOR i=1 TO 2:FOR k=1 TO 2
160 PRINT"a("i","k")=";:INPUT a(i,k)
170 b(i,k)=a(i,k)
180 NEXT k:PRINT:NEXT i
190 ' Eingabekontrolle
200 FOR i=1 TO 2:FOR k=1 TO 2:PRINT a(i,k);:NEXT k
210 PRINT:NEXT i
220 PRINT:PRINT "Eingabe in Ordnung (j/n)"
230 e$=INKEY$:IF e$="" THEN 230
240 PRINT:IF e$="n" THEN CLS:GOTO 70
250 RETURN
260 '
270 ' Berechnung der Koeff. der char. Gleichung
280 p(1)=-(a(1,1)+a(2,2))
```

```
290 p(0)=a(1,1)*a(2,2)-a(1,2)*a(2,1):p(2)=1:RETURN
300 '
310 'Loesung der char. Gleichung
320 s=-p(1)/2:w=s*s-p(0)
330 IF ABS(w)<10↑(-6) THEN w=0
340 v=SQR(ABS(w)):IF w<0 THEN RETURN
350 z(1)=s-v:z(2)=s+v:RETURN
360 '
370 ' Ausgabe der Koeff. der char. Gleichung
380 PRINT"Koeffizienten der char.Gleichung :"
390 FOR k=2 TO 0 STEP -1
400 PRINT"p("k")="p(k):NEXT k:PRINT:RETURN
410 '
420 ' Ausgabe der Eigenwerte
430 IF w<0 THEN 490
440 IF w>0 THEN 470
450 PRINT"Ein doppelter reeller Eigenwert :"
460 PRINT z(1):RETURN
470 PRINT"2 reelle Eigenwerte :"
480 FOR k=1 TO 2:PRINT z(k),:NEXT :PRINT:RETURN
490 PRINT s"-i*"v
500 PRINT s"+i*"v" sind kompl.Eigenwerte":END
510 '
520 ' Berechnung eines EV zum EW x
530 FOR k=0 TO 2:FOR i=0 TO 2
540 y(i,k)=0:b(i,k)=a(i,k):NEXT i:NEXT k
550 FOR i=1 TO 2:b(i,i)=a(i,i)-x:NEXT i
560 te=b(1,1)↑2+b(2,1)↑2+b(1,2)↑2+b(2,2)↑2
570 IF te=0 THEN r=2:GOTO 620
580 IF b(1,1)=0 THEN IF b(2,1)=0 THEN r=1:GOTO 630
590 IF b(1,1)=0 THEN r=1:GOTO 640
600 b(2,2)=b(2,2)-b(1,2)*b(2,1)/b(1,1)
610 r=1:GOTO 650
620 y(1,1)=1:y(2,1)=0:y(1,2)=0:y(2,2)=1:GOTO 680
630 y(1,1)=1:y(2,1)=0:GOTO 680
640 y(1,1)=-b(2,2):y(2,1)=a(2,1):GOTO 680
650 y(1,1)=-b(1,2):y(2,1)=b(1,1)
```

```
660 '
670 ' Ausdruck der Eigenvektoren
680 PRINT:PRINT"Der zu"x"gehoerende Eigenraum"
690 PRINT"hat die Dimension"r;:PRINT".Basis ist :"
700 FOR i=1 TO r:PRINT"(";
710 PRINT y(1,i)"/"y(2,i)")":NEXT i:RETURN
```

Beispiel 1

$$A = \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}$$

charakteristische Gleichung : $x^2 - 6x + 5 = 0$

Eigenwert 1 , zugeh. Eigenvektor (-1/1)

Eigenwert 5 , zugeh. Eigenvektor (-1/-1)

Beispiel 2

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

charakteristische Gleichung : $x^2 - 2x + 1 = 0$

doppelter Eigenwert $x_{1,2} = 1$

Der zugehörige Eigenraum ist eindimensional,Eigenvektor (1/0)

Beispiel 3

$$A = \begin{pmatrix} 3 & -5 \\ 2 & 1 \end{pmatrix}$$

charakteristische Gleichung : $x^2 - 4x + 13 = 0$

Eigenwerte $x_{1,2} = 2 \pm 3i$

Zugehörige Eigenvektoren können mit dem Programm 4 dieses Abschnitts ermittelt werden :

Eigenvektor zu 2-3i : (1-3i/2)

Eigenvektor zu 2+3i : (1+3i/2).

2. Eigenwertproblem für 3 × 3 Matrizen

```
10 CLS
20 PRINT"Eigenwertaufgabe fuer 3*3-Matrizen":PRINT
30 DIM d(4),e(2),a(3,3),g(3,3)
40 DIM q(3,3),c(3)
50 '
60 ' Hauptprogramm
70 GOSUB 160:GOSUB 290:GOSUB 590
80 GOSUB 400:GOSUB 1170:e(0)=x:GOSUB 490:GOSUB 650
90 GOSUB 740:x=e(0):GOSUB 810:IF f<2 THEN 130
100 x=e(2):IF d2<d THEN 120
110 GOSUB 810:IF f<3 THEN 130
120 x=e(1):GOSUB 810
130 END
140 '
150 ' Koeffizienteneingabe
160 FOR i=1 TO 3:FOR k=1 TO 3
170 PRINT"a("i","k")=";:INPUT a(i,k)
180 g(i,k)=a(i,k)
190 NEXT k:PRINT:NEXT i
200 ' Eingabekontrolle
210 FOR i=1 TO 3:FOR k=1 TO 3:PRINT a(i,k);:NEXT k
220 PRINT:NEXT i
230 PRINT:PRINT"Eingabe in Ordnung (j/n)"
240 e$=INKEY$:IF e$="" THEN 240
250 PRINT:IF e$="n" THEN CLS:GOTO 70
260 RETURN
270 '
280 ' Berechnung der Koeff. der char. Gleichung
290 h=a(1,1)*a(2,2)*a(3,3)+a(1,2)*a(2,3)*a(3,1)
    +a(1,3)*a(2,1)*a(3,2)
300 i=a(1,3)*a(2,2)*a(3,1)+a(1,2)*a(2,1)*a(3,3)
    +a(1,1)*a(2,3)*a(3,2)
310 d(0)=h-i
320 h=a(1,2)*a(2,1)+a(1,3)*a(3,1)-a(1,1)*a(2,2)-a(1,1)*a(3,3)
330 d(1)=h-a(2,2)*a(3,3)+a(2,3)*a(3,2)
340 d(2)=a(1,1)+a(2,2)+a(3,3)
```

```
350 d(3)=-1
360 RETURN
370 '
380 ' Loesung der char.Gleichung
390 ' 1.Lsg. durch Naeherungsverfahren
400 x=0
410 s=SGN(-d(0)):b=10
420 b=b/10:s=-s:IF ABS(b)<0.000000001 THEN RETURN
430 x=x+b*s
440 f=d(0)+x*(d(1)+x*(d(2)-x))
450 IF f=0 THEN RETURN
460 IF f*s>0 THEN 430
470 GOTO 420
480 ' Reduktion auf quadr. Gleichung
490 FOR i=3 TO 0 STEP -1
500 d(i)=x*d(i+1)+d(i):NEXT i
510 ' Loesung der quadr. Gleichung
520 s=d(2)/2:w=s*s+d(1)
530 IF ABS(w)<10↑(-6) THEN w=0
540 v=SQR(ABS(w)):IF w<0 THEN 560
550 x=s-v:GOSUB 1170:e(1)=x:x=s+v:GOSUB 1170:e(2)=x
560 RETURN
570 '
580 ' Ausgabe der Koeff.der char.Gleichung
590 PRINT"Koeffizienten der char.Gleichung :"
600 FOR k=3 TO 0 STEP -1
610 PRINT"d("k")="d(k):NEXT k:PRINT
620 RETURN
630 '
640 ' Ausgabe der Eigenwerte
650 IF w<0 THEN 680
660 PRINT"3 reelle Eigenwerte :"
670 FOR k=0 TO 2:PRINT e(k),:NEXT:PRINT:RETURN
680 PRINT e(0)" ist reeller Eigenwert"
690 PRINT s"-i*"v
700 PRINT s"+i*"v" sind kompl.Eigenwerte"
710 RETURN
```

```
720 '
730 ' Art u. Vielfachheit der Eigenwerte
740 IF w<0 THEN f=0:RETURN
750 d=0.0000001:d1=ABS(e(0)-e(1)):d2=ABS(e(0)-e(2))
760 IF d1<d AND v=0 THEN f=1:RETURN
770 IF d1<d OR v=0 OR d2<d THEN f=2:RETURN
780 f=3:RETURN
790 '
800 ' Berechnung der Eigenvektoren
810 FOR k=0 TO 3:FOR i=0 TO 3
820 q(i,k)=0:g(i,k)=a(i,k):NEXT i:NEXT k
830 FOR i=1 TO 3:g(i,i)=a(i,i)-x:NEXT i
840 'Loesung des hom. LGS
850 f0=0
860 FOR k=1 TO 3
870 FOR i=1 TO 3
880 IF ABS(g(i,k))<10↑(-5) THEN g(i,k)=0
890 IF (g(i,0)>0 OR g(i,k)=0) THEN NEXT i:GOTO 980
900 g(0,k)=i:g(i,0)=k:f0=f0+1
910 FOR l=1 TO 3:c(l)=g(i,l):NEXT l
920 FOR j=1 TO 3
930 y=g(j,k)/c(k)
940 FOR l=1 TO 3
950 g(j,l)=g(j,l)-c(l)*y
960 NEXT l:NEXT j
970 FOR l=1 TO 3:g(i,l)=c(l)/c(k):NEXT l
980 NEXT k
990 v=0:w=0:r=3-f0
1000 FOR k=1 TO 3
1010 i=g(0,k)
1020 IF i<1 THEN v=v+1:q(k,v)=1:GOTO 1070
1030 FOR l=1 TO 3
1040 IF g(0,l)>0 THEN 1060
1050 w=w+1:q(k,w)=-g(i,l)
1060 NEXT l:w=0
1070 NEXT k
1080 ' Ausdruck der Eigenvektoren
```

```
1090 PRINT:PRINT"Der zu "x"gehoerende Eigenraum"
1100 PRINT"hat die Dimension"r;:PRINT".Eine Basis ist :"
1110 FOR i=1 TO r:PRINT"(";
1120 FOR p=1 TO 3:x=q(p,i):GOSUB 1170:q(p,i)=x:NEXT p
1130 PRINT q(1,i)"/"q(2,i)"/"q(3,i)")":NEXT i
1140 RETURN
1150 '
1160 ' Rundung
1170 x=1000*x:IF x-INT(x)<10↑(-4) THEN x=INT(x):GOTO 1190
1180 IF INT(x)+1-x<10↑(-4) THEN x=INT(x)+1
1190 x=x/1000:RETURN
```

Beispiel 1

$$A = \begin{pmatrix} 4 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{pmatrix}$$

charakteristische Gleichung : $-x^3 + 5x^2 - 4x = 0$

Eigenwert 0 , zugeh. Eigenvektor $(\frac{1}{4}/-2/1)$

Eigenwert 4 , zugeh. Eigenvektor (1/0/0)

Eigenwert 1 , zugeh. Eigenvektor $(-\frac{2}{3}/1/0)$

Beispiel 2

$$A = \begin{pmatrix} 4 & 5 & 6 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

charakteristische Gleichung : $-x^3 + 8x^2 - 20x + 16 = 0$

Eigenwert 2 (doppelt) , zugeh. Eigenvektor (-2,5/1/0)

Eigenwert 4 , zugeh. Eigenvektor (1/0/0).

Beispiel 3

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

charakteristische Gleichung : $-x^3 + x^2 - x + 1 = 0$

Eigenwert 1 , zugeh. Eigenvektor (1/0/0)

Die zu den komplexen Eigenwerten $\pm$ i zugehörigen Eigenvektoren kann man mit dem Programm 4 bestimmen.

Eigenwert i , zugeh. Eigenvektor (0/i/1)

Eigenwert -i , zugeh. Eigenvektor (0/-i/1).

3.Bestimmung des charakteristischen Polynoms

Dieses Programm ermittelt die Koeffizienten des charakteristischen Polynoms p(x) = |A - xE| einer reellen n x n-Matrix A auf folgende Weise :
Für n+1 verschiedene x-Werte wird die Determinante |A - xE| berechnet.Dann wird das durch diese n+1 Punkte bestimmte Interpolationspolynom bestimmt.

```
10 CLS:PRINT "charakteristisches Polynom":PRINT
20 INPUT "Zeilenzahl :";n
30 DIM a(n,n),b(n,n),c(n+1,n+1),ce(n)
40 DIM x(n),y(n),z(n),d(n),p(n)
50 '
60 ' Koeffizienteneingabe
70 PRINT:PRINT "Koeffizienteneingabe zeilenweise":PRINT
80 FOR i=1 TO n:FOR k=1 TO n
90 PRINT "a("i","k")";:INPUT b(i,k):NEXT k:PRINT
100 NEXT i
110 ' Eingabekontrolle
120 CLS:PRINT"Koeffizientenschema":PRINT
130 GOSUB 490:PRINT
140 PRINT"Eingabe korrekt (j/n)"
150 e$=INKEY$:IF e$="" THEN 150
160 PRINT:IF e$="n" THEN 80
170 '
180 m=INT(-n/2)
190 FOR e=m TO m+n
200 FOR i=1 TO n:FOR k=1 TO n:a(i,k)=b(i,k):NEXT:NEXT
210 FOR i=1 TO n:a(i,0)=0:a(i,i)=b(i,i)-e:NEXT i
220 d=1:s=1:f=0
230 GOSUB 300
240 x(r)=e:y(r)=x:r=r+1
250 NEXT e
260 GOSUB 670
270 END
280 '
290 ' Eliminationsschritte
```

```
300 FOR k=1 TO n
310 FOR i=1 TO n
320 IF ABS(a(i,k))<10↑(-5) THEN a(i,k)=0
330 IF (a(i,0)>0 OR a(i,k)=0) THEN 420
340 a(0,k)=i:a(i,0)=k:f=f+1
350 FOR l=1 TO n:ce(l)=a(i,l):NEXT l
360 d=d*ce(k)
370 FOR j=i TO n
380 y=a(j,k)/ce(k)
390 FOR l=1 TO n
400 a(j,l)=a(j,l)-ce(l)*y
410 NEXT l:NEXT j
420 NEXT i
430 NEXT k
440 IF f<n THEN d=0
450 GOSUB 600:x=d*s
460 RETURN
470 '
480 ' Ausdruck Koeffizientenschema
490 FOR p=1 TO n:FOR q=1 TO n
500 PRINT b(p,q);:NEXT q:PRINT
510 NEXT p
520 RETURN
530 '
540 ' Runden
550 IF x-INT(x)<10↑(-4) THEN x=INT(x):GOTO 570
560 IF INT(x)+1-x<10↑(-4) THEN x=INT(x)+1
570 RETURN
580 '
590 ' Fehlstaende
600 FOR p=1 TO n-1:FOR q=p+1 TO n
610 IF a(0,p)<a(0,q) THEN 630
620 s=-s
630 NEXT q:NEXT p
640 RETURN
650 '
660 ' Dividierte Differenzen
```

```
670 c(1,1)=1
680 z(0)=y(0):d(0)=y(0)
690 FOR j=1 TO n:d(j)=y(j)
700 FOR k=j-1 TO 0 STEP -1
710 d(k)=(d(k+1)-d(k))/(x(j)-x(k))
720 NEXT k
730 z(j)=d(0)
740 NEXT j:p(0)=z(0)
750 '
760 ' Koeff. des Interpol.polynoms
770 FOR i=2 TO n+1:FOR k=1 TO i
780 c(i,k)=c(i-1,k-1)-c(i-1,k)*x(i-2)
790 p(k-1)=p(k-1)+c(i,k)*z(i-1)
800 NEXT k:NEXT i
810 '
820 ' Ausgabe der Koeffizienten
830 PRINT:PRINT"Koeffizienten :":PRINT
840 FOR k=n TO 0 STEP -1
850 x=p(k):GOSUB 550:PRINT"p("k")="x:NEXT k
860 RETURN
```

Beispiel : Bestimmen Sie das char. Polynom von

$$\begin{pmatrix} -2 & -2 & 6 & -5 & 4 & 1 \\ -6 & -6 & 14 & -13 & 11 & 2 \\ -10 & -11 & 23 & -19 & 16 & 3 \\ -10 & -13 & 25 & -18 & 13 & 6 \\ -6 & -13 & 21 & -17 & 7 & 11 \\ -6 & -13 & 21 & -17 & 1 & 17 \end{pmatrix}$$

Nach knapp einer halben Minute (SCHNEIDER CPC) hat man :

$p(x) = x^6 - 21x^5 + 175x^4 - 735x^3 + 1624x^2 - 1764x + 720.$

Mit dem Programm 'Nullst.3' erhält man nun (ca. 2 Minuten) die Eigenwerte : $x_1=1$, $x_2=2$, $x_3=3$, $x_4=4$, $x_5=5$, $x_6=6$

Beispiel : Geben Sie die Nullmatrix und die Einheitsmatrix für ein größeres n ein.

Bei n=10 geht es noch gut , bei n=15 gibt es Unsinn.

Etwas Mißtrauen ist nach aufwendiger Rechnung stets geboten !

4. Eigenvektoren zu bekannten Eigenwerten

Für eine reelle Matrix der Zeilenzahl n und einen bekannten reellen oder komplexen Eigenwert bestimmt dieses Programm die Dimension und eine Basis des zugehörigen Eigenraums.

```
10 CLS:PRINT"Eigenvektoren zu komplexen Eigenwerten"
20 PRINT"bei reellen Matrizen":PRINT
30 INPUT "Zeilenzahl";n
40 DIM a1(n,n):DIM a2(n,n):DIM c1(n):DIM c2(n)
50 DIM z1(n,n):DIM z2(n,n)
60 g=10^(-7)
70 '
80 ' Koeffizienteneingabe
90 PRINT"Koeffizienteneingabe zeilenweise":PRINT
100 FOR i=1 TO n:FOR k=1 TO n
110 PRINT"a("i","k") :";:INPUT a1(i,k)
120 NEXT k:PRINT:NEXT i
130 INPUT "Eigenwert : Re,Im";u,v
140 FOR i=1 TO n:a1(i,i)=a1(i,i)-u:a2(i,i)=-v:NEXT i
150 '
160 'Eliminationsschritte
170 FOR k=1 TO n
180 FOR i=1 TO n
190 IF (a(i,0)>0 OR (a1(i,k)=0 AND a2(i,k)=0)) THEN NEXT i:GOTO 350
200 a(0,k)=i:a(i,0)=k:f=f+1
210 FOR l=1 TO n:c1(l)=a1(i,l):c2(l)=a2(i,l):NEXT l
220 FOR j=1 TO n:ne=c1(k)*c1(k)+c2(k)*c2(k)
230 y1=(a1(j,k)*c1(k)+a2(j,k)*c2(k))/ne
240 y2=(a2(j,k)*c1(k)-a1(j,k)*c2(k))/ne
250 FOR l=1 TO n
260 a1(j,l)=a1(j,l)-(c1(l)*y1-c2(l)*y2)
270 IF ABS(a1(j,l))<g THEN a1(j,l)=0
280 a2(j,l)=a2(j,l)-(c1(l)*y2+c2(l)*y1)
290 IF ABS(a2(j,l))<g THEN a2(j,l)=0
300 NEXT l:NEXT j
310 FOR l=1 TO n
320 a1(i,l)=(c1(l)*c1(k)+c2(l)*c2(k))/ne
330 a2(i,l)=(c2(l)*c1(k)-c1(l)*c2(k))/ne
```

```
340 NEXT l
350 NEXT k
360 '
370 ' Berechnung der Loesung
380 r=n-f:v=0
390 FOR k=1 TO n
400 i=a(0,k)
410 IF i<1 THEN v=v+1:z1(k,v)=1:GOTO 460
420 FOR l=1 TO n
430 IF a(0,l)>0 THEN 450
440 w=w+1:z1(k,w)=-a1(i,l):z2(k,w)=-a2(i,l)
450 NEXT l:w=0
460 NEXT k
470 '
480 ' Ausgabe der Loesung
490 PRINT:PRINT"Der Eigenraum hat die Dimension"r
500 PRINT"Eine Basis ist:"
510 FOR i=1 TO r:PRINT"[";
520 FOR k=1 TO n
530 x=z1(k,i):GOSUB 600:u=x
540 x=z2(k,i):GOSUB 600:v=x
550 IF k<n THEN PRINT "("u"/"v")""/";:NEXT k
560 PRINT "("u"/"v")]":NEXT i
570 END
580 '
590 ' Runden
600 x=1000*x:IF x-INT(x)<10↑(-4) THEN x=INT(x):GOTO 620
610 IF INT(x)+1-x<10↑(-4) THEN x=INT(x)+1
620 x=x/1000:RETURN
```

Beispiel Die nachfolgend angegebene Matrix besitzt den Eigenwert i. Bestimmen Sie eine Basis des zugehörigen Eigenraums.

$$\begin{pmatrix} 1 & 0 & -2 & 0 \\ 0 & 2 & 0 & 0 \\ 1 & 0 & -1 & 1 \\ 0 & 3 & 0 & 2 \end{pmatrix}$$

Der Eigenraum ist eindimensional, ein Eigenvektor ist

(1+i/0/1/0)

5. Iterationsverfahren von MISES

Zur Bestimmung eines Eigenwerts und Eigenvektors einer reellen Matrix A verwendet dieses Verfahren im Wesentlichen die Iteration

$\vec{x}_0$ ein beliebiger Startvektor

$\vec{x}_{n+1} = A \cdot \vec{x}_n$ für $n \in \mathbb{N}$

Die Koordinaten von $\vec{x}_n$ würden dabei jedoch unangenehm groß werden. Man rechnet stattdessen mit

$$\vec{x}_{n+1} = A \cdot \vec{y} \ ,$$

wobei sich $\vec{y}$ aus $\vec{x}_n$ durch Division durch die maximale Koordinate ergibt.

Folgendes kann passieren :

1) Die Matrix A besitzt genau einen reellen , dominanten Eigenwert x_1. Für die Eigenwerte $x_1, x_2, ..$ gelte

$$|x_1| > |x_2| \geq |x_3| \geq ..$$

Das Verfahren konvergiert.
Im Allgemeinen ergibt sich der dominante Eigenwert und ein zugehöriger Eigenvektor.
Für spezielle Startvektoren kann das Verfahren auch andere Eigenwerte und Eigenvektoren liefern. s.Beispiel 1

2) Der betragsgrößte Eigenwert ist mehrfacher Eigenwert. Das Verfahren konvergiert, meist jedoch sehr langsam. s.Beispiel 2

3) Das Verfahren konvergiert nicht allgemein, sondern nur für spezielle Startwerte. s.Beispiel 3

Um die Konvergenz verfolgen zu können, werden in Zeile 110 die Zahlen m und d ausgegeben. Dabei sind :

- m Näherung für den Eigenwert
- d Summe der Abweichungen der Koordinaten von $\vec{x}_n$ von den Koordinaten von $\vec{x}_{n-1}$.

```
10 CLS:PRINT"v.Mises-Verfahren zur Bestimmung eines"
20 PRINT"Eigenwerts und eines Eigenvektors einer Matrix":PRINT
30 INPUT "n=";n
40 DIM a(n,n),x(n),y(n),z(n)
50 '
60 ' Hauptprogramm
```

```
70 GOSUB 250:´ Matrixeingabe
80 GOSUB 360:´ Startvektor
90 GOSUB 510:´ m
95 IF m=0 THEN FOR k=1 TO n:x(k)=ma*z(k):NEXT :GOTO 180
100 d=0:FOR k=1 TO n:x(k)=x(k)/m:d=d+ABS(z(k)-x(k)):NEXT k
110 PRINT m,d
120 IF d<0.00000001 THEN 180
130 FOR k=1 TO n:z(k)=x(k):NEXT k
140 GOSUB 430:´ y=A*x
150 FOR k=1 TO n:x(k)=y(k):NEXT k
160 e=e+1:ma=m:GOTO 90
170 ´ Loesung
180 PRINT:x=m:GOSUB 590:PRINT"Ein Eigenwert :"x
190 PRINT:PRINT"Ein Eigenvektor :":PRINT
200 FOR k=1 TO n:x=x(k):GOSUB 590:PRINT"x("k")="x:NEXT k:PRINT
210 PRINT"Anzahl der Iterationen"e
220 END
230 ´
240 ´ Koeffizienteneingabe
250 FOR i=1 TO n:FOR k=1 TO n
260 PRINT"a("i","k")";:INPUT a(i,k)
270 NEXT k:PRINT:NEXT i
280 ´ Eingabekontrolle
290 FOR i=1 TO n:FOR k=1 TO n:PRINT a(i,k);:NEXT k:PRINT:NEXT i
300 PRINT:PRINT"Eingabe korrekt (j/n)"
310 e$=INKEY$:IF e$="" THEN 310
320 IF e$="n" THEN 250
330 RETURN
340 ´
350 ´ Startvektor
360 PRINT:PRINT"Startvektor :":PRINT
370 FOR k=1 TO n
380 PRINT"x("k")";:INPUT x(k)
390 NEXT k:PRINT
400 RETURN
410 ´
420 ´ y=A*x
430 FOR i=1 TO n:s=0
```

```
440 FOR k=1 TO n
450 s=s+a(i,k)*x(k)
460 NEXT k:y(i)=s
470 NEXT i
480 RETURN
490 '
500 ' Betragsgr.Komponente
510 m=ABS(x(1)):j=1
520 FOR k=2 TO n:x=ABS(x(k))
530 IF x>m THEN m=x:j=k
540 NEXT k
550 m=x(j)
560 RETURN
570 '
580 ' Runden
590 x=x*1000:IF x-INT(x)<10↑(-4) THEN x=INT(x):GOTO 610
600 IF INT(x)+1-x<10↑(-4) THEN x=INT(x)+1
610 x=x/1000:RETURN
```

Beispiel 1

$$A = \begin{pmatrix} 4 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{pmatrix}$$

Die Matrix hat offensichtlich die Eigenwerte 4,1,0.

a) Startvektor (1/1/1)
   Das Verfahren konvergiert gegen den dominanten Eigenwert 4 und den zugehörigen Eigenvektor (1/0/0).

b) Startvektor (-1/-5/4)
   Das Verfahren konvergiert gegen den Eigenwert 1 mit zugehörigem Eigenvektor ($-\frac{2}{3}$/1/0).
   (Erklärung : Der Startvektor ist eine Linearkombination von Eigenvektoren zu den Eigenwerten 0 und 1)

Beispiel 2

$$A = \begin{pmatrix} 2 & 3 & 4 \\ 0 & 2 & 5 \\ 0 & 0 & 1 \end{pmatrix}$$

2 ist doppelter, 1 ist einfacher Eigenwert der Matrix.

a) Die Startvektoren (1/0/0) oder (11/-5/1) führen unmittelbar zu einem Ergebnis : es sind nämlich Eigenvektoren zu den Eigenwerten 2 bzw. 1.

b) Startvektor (0/-5/1)
Nach etwa 25 Schritten ist man beim Eigenwert 2.
(Der Startvektor ist eine Linearkombination von Eigenvektoren).

c) Startvektor (1/1/1)
Das Verfahren konvergiert, allerdings sehr, sehr langsam.

Beispiel 3

$$A = \begin{pmatrix} 1 & 1 \\ 3 & -1 \end{pmatrix}$$

Eigenwerte sind 2 und -2.

a) Startvektor (1/0) : Divergenz

b) Startvektor (1/2) : Divergenz

c) Startvektor (1/1) : Der Startvektor ist bereits ein Eigenvektor zum Eigenwert 2.

Hinweis : Soll bei einer Matrix das Verfahren mit unterschiedlichen Startvektoren ausprobiert werden, programmiert man :

```
220 e=0:GOTO 80
```

(Dadurch erspart man sich die Neueingabe der Matrix)

Vektorrechnung*

Dieses Programm enthält folgende Unterprogramme :

1 Punkt - Punkt

Berechnung des Abstands zweier Punkte im Raum.

2 Punkt - Gerade

Berechnung des Abstands des Punktes von der Geraden und ggf. Bestimmung des Lotfußpunktes.

3 Punkt - Ebene

Berechnung des Abstands des Punktes von der Ebene und ggf. Bestimmung des Lotfußpunktes.

4 Gerade - Gerade

Bei windschiefen Geraden : Berechnung der Punkte minimaler Entfernung und Abstand der Geraden.
Bei sich schneidenden Geraden : Berechnung von Schnittpunkt und Schnittwinkel.

5 Gerade - Ebene

Sind Gerade und Ebene parallel, wird ihr Abstand angegeben, andernfalls werden Durchstoßpunkt und Schnittwinkel berechnet.

6 Ebene - Ebene

Bei parallelen Ebenen wird der Abstand berechnet. Bei nicht-parallelen Ebenen werden ein Punkt und die Richtung der Schnittgeraden sowie der Schnittwinkel angegeben.

Die Koordinaten von Vektoren oder Punkten (x,y,z) sind einzugeben in der Form : x,y,z RETURN

```
10 CLS:PRINT"Vektorrechnung :":PRINT
20 PRINT"Punkte,Geraden und Ebenen"
30 PRINT"(Winkel,Abstaende,Lotfusspunkte"
40 PRINT" Schnittpunkte,Schnittgeraden)":PRINT
50 PRINT"1 Punkt-Punkt"
60 PRINT"2 Punkt-Gerade"
70 PRINT"3 Punkt-Ebene"
```

*) gegebenenfalls PI = 4*ATN(1) definieren!

```
80 PRINT"4 Gerade-Gerade"
90 PRINT"5 Gerade-Ebene"
100 PRINT"6 Ebene-Ebene":PRINT
110 PRINT"7 Ende":PRINT
120 PRINT"Welches Programm"
130 e$=INKEY$:IF e$="" THEN 130
140 e=VAL(e$):ON e GOSUB 180,240,330,400,690,800,1470
150 e$=INKEY$:IF e$="" THEN 150
160 CLS:GOTO 50
170 '
180 CLS:PRINT"Abstand Punkt-Punkt":PRINT:z$=""
190 GOSUB 960:GOSUB 1000:u=x:v=y:w=z
200 GOSUB 960:GOSUB 1000
210 GOSUB 1020
220 GOSUB 1040:RETURN
230 '
240 CLS:PRINT"Punkt-Gerade":PRINT:z$=""
250 GOSUB 960:GOSUB 1000:u=x:v=y:w=z
260 z$=" der Geraden"
270 GOSUB 960:GOSUB 1000:a=x:b=y:c=z
280 GOSUB 980:GOSUB 1000
290 t=((u-a)*x+(v-b)*y+(w-c)*z)/(x*x+y*y+z*z)
300 x=a+t*x:y=b+t*y:z=c+t*z
310 GOSUB 1020:GOSUB 1040:GOSUB 1060:RETURN
320 '
330 CLS:PRINT"Punkt-Ebene":PRINT:z$=""
340 GOSUB 960:GOSUB 1000:u=x:v=y:w=z
350 GOSUB 1170
360 GOSUB 1210
370 GOSUB 1060
380 RETURN
390 '
400 CLS:PRINT"Gerade - Gerade":PRINT:z$="":p=0
410 z$=" der 1.Geraden"
420 GOSUB 960:GOSUB 1000:a1=x:a2=y:a3=z
430 GOSUB 980:GOSUB 1000
440 b1=x:b2=y:b3=z:a=x↑2+y↑2+z↑2
```

```
450 z$=" der 2.Geraden":GOSUB 960:GOSUB 1000
460 a4=x:a5=y:a6=z
470 c1=x-a1:c2=y-a2:c3=z-a3
480 e=c1*b1+c2*b2+c3*b3
490 GOSUB 980:GOSUB 1000
500 d=-(x↑2+y↑2+z↑2)
510 b=-(x*b1+y*b2+z*b3)
520 co=-b/SQR(-a*d)
530 f=c1*x+c2*y+c3*z
540 k=a*d+b*b:IF ABS(k)<10↑(-7) THEN k=0
550 IF k=0 THEN GOSUB 1100:GOTO 570
560 s=(e*b+a*f)/k:t=(e-b*s)/a:GOTO 570
570 x=a4+s*x:y=a5+s*y:z=a6+s*z
580 u=a1+t*b1:v=a2+t*b2:w=a3+t*b3
590 GOSUB 1020
600 IF ABS(d)<10↑(-7) THEN d=0
610 PRINT:PRINT"Abstand"d:IF p=1 THEN 670
620 IF d=0 THEN PRINT:GOSUB 1120:GOTO 670
630 PRINT:PRINT"Minimalpunkt auf 1.Geraden :"
640 PRINT"("u"/"v"/"w")"
650 PRINT:PRINT"Minimalpunkt auf 2.Geraden :"
660 PRINT"("x"/"y"/"z")"
670 RETURN
680 '
690 CLS:PRINT"Gerade - Ebene"
700 z$=" der Geraden":GOSUB 960:GOSUB 1000
710 u=x:v=y:w=z
720 GOSUB 980:GOSUB 1000
730 b1=x:b2=y:b3=z
740 GOSUB 1170
750 s=b1*x+b2*y+b3*z:b=x↑2+y↑2+z↑2
760 IF s=0 THEN GOSUB 1250:GOTO 780
770 GOSUB 1280
780 RETURN
790 '
800 CLS:PRINT"Ebene - Ebene":PRINT
810 PRINT:PRINT"1.Ebene :":GOSUB 1170
```

```
820 u=x:v=y:w=z:e=d:n=SQR(x↑2+y↑2+z↑2)
830 PRINT:PRINT"2.Ebene :":GOSUB 1170
840 a=v*z-w*y:b=w*x-u*z:c=u*y-v*x
850 m=SQR(x↑2+y↑2+z↑2):co=(u*x+v*y+w*z)/n/m
860 IF c<>0 THEN GOSUB 1330:GOTO 900
870 IF b<>0 THEN GOSUB 1350:GOTO 900
880 IF a<>0 THEN GOSUB 1370:GOTO 900
890 GOSUB 1400:GOTO 940
900 PRINT:PRINT"Ein Punkt der Schnittgeraden :"
910 PRINT"("x1"/"x2"/"x3")"
920 PRINT"Richtung der Schnittgeraden :":PRINT"("a"/"b"/"c")"
930 GOSUB 1130
940 RETURN
950 '
960 PRINT:PRINT"Punkt"z$" eingeben":RETURN
970 '
980 PRINT:PRINT"Richtung"z$" eingeben":RETURN
990 '
1000 INPUT x,y,z:RETURN
1010 '
1020 d=SQR((u-x)↑2+(v-y)↑2+(w-z)↑2):RETURN
1030 '
1040 PRINT:PRINT"Abstand";d:RETURN
1050 '
1060 IF d=0 THEN RETURN
1070 PRINT:PRINT"Lotfusspunkt :":PRINT
1080 PRINT"("x"/"y"/"z")":PRINT:RETURN
1090 '
1100 PRINT:PRINT"parallele Geraden":p=1:s=1:t=(e-b)/a:RETURN
1110 '
1120 PRINT:PRINT"Schnittpunkt :("x"/"y"/"z")"
1130 IF co=0 THEN w=90:GOTO 1150
1140 w=(ATN(SQR(1-co↑2)/co))*180/PI
1150 PRINT"Schnittwinkel :"ABS(w)" Grad":RETURN
1160 '
1170 PRINT:PRINT"Normalenvektor der Ebene :"
1180 GOSUB 1000
```

```
1190 PRINT:PRINT"Rechte Seite der Ebenengleichung :";
     INPUT d:RETURN
1200 '
1210 t=(d-(u*x+v*y+w*z))/(x*x+y*y+z*z)
1220 x=u+t*x:y=v+t*y:z=w+t*z
1230 GOSUB 1020:GOSUB 1040:RETURN
1240 '
1250 PRINT:PRINT"Gerade und Ebene sind parallel"
1260 GOSUB 1210:RETURN
1270 '
1280 t=(d-(u*x+v*y+w*z))/s
1290 x=u+t*b1:y=v+t*b2:z=w+t*b3
1300 co=SQR(1-s*s/(b1↑2+b2↑2+b3↑2)/b)
1310 GOSUB 1120:RETURN
1320 '
1330 x3=1:e=e-w:d=d-z:x1=(e*y-d*v)/c
1340 x2=(u*d-x*e)/c:RETURN
1350 x2=1:e=e-v:d=d-y:x1=(-e*z+d*w)/b
1360 x3=(-u*d+x*e)/b:RETURN
1370 x1=1:e=e-u:d=d-x:x2=(e*z-d*w)/a
1380 x3=(v*d-y*e)/a:RETURN
1390 '
1400 PRINT:PRINT"Die Ebenen sind parallel"
1410 IF SGN(x)=SGN(u) AND SGN(y)=SGN(v) AND
     SGN(z)=SGN(w) THEN 1430
1420 d=-d
1430 d=ABS(e/n-d/m)
1440 IF ABS(d)<10↑(-7) THEN d=0
1450 PRINT"Abstand"d:RETURN
1460 '
1470 END
```

Beispiel 1

P = (1/3/-4) Q = (2/3/1) Abstand : 5.099

Beispiel 2

P = (1/2/3) g: $\vec{x}$ = (1/1/0) + t(-1/2/-1)

Abstand : 3.1358 Lotfußpunkt (1.1667/0.6667/0.1667)

Beispiel 3

P = (1/2/3) E : 2x - y + z = 4

Normalenvektor von E ist also (2/-1/1).

Abstand : 0.4082 Lotfußpunkt (1.3333/1.8333/3.1666)

Beispiel 4

a) g_1:$\vec{x}$=(1/2/1)+t(1/0/1) , g_2:$\vec{x}$=(1/1/1)+s(1/-1/0)

windschiefe Geraden : Abstand 0.577 Minimalpunkte :

P_1=(0.6666/2/0.6666) sowie P_2=(0.3333/1.6666/1)

b) g_1:$\vec{x}$=(2/-1/3)+t(1/4/0) , g_2:$\vec{x}$=(2/1/1)+s(1/2/2)

Abstand 0 , Schnittpunkt (3/3/3) , Schnittwinkel 43.3°

c) g_1:x=(1/1/0)+t(-2/4/-8) , g_2:x=(3/2/1)+s(1/-2/4)

parallele Geraden , Abstand 2.288

Beispiel 5

g : $\vec{x}$=(1/0/1)+t(2/1/0) , E : 2x + y - 2z = 1

Schnittpunkt (1.4/0.2/1) Schnittwinkel 48.19°

Beispiel 6

a) E_1 : 2x - y + 3z = 2 , E_2 : x + y - 3z = 1

(1/3/1) ist Punkt der Schnittgeraden , (0/9/3) ihre Richtung. Schnittwinkel : 49.86°

Schnittgerade : $\vec{x}$ = (1/3/1) + t(0/3/1)

b) E_1 : -4x + 2y - 6z = 7 , E_2 : 2x - y + 3z = 10

parallele Ebenen , Abstand 3.608

Rechnen mit komplexen Zahlen*

Das Rechnen mit komplexen Zahlen gestaltet sich bei diesem Programm ähnlich einfach wie das Rechnen mit reellen Zahlen auf dem Taschenrechner.Insbesondere sind Kettenrechnungen wie

$$\frac{(1 + 2i)^3 - (3 - 4i)}{6 + 2i}$$

ohne Umstände durchzuführen.

Dies wird ermöglicht, indem die Verarbeitung der Zahlen nach der 'Umgekehrten polnischen Notation' erfolgt.(s.Programm Bruchrechnung) :

Eingegebene komplexe Zahlen werden auf einem 'Stapel' abgelegt und eine nachfolgende Rechenoperation verknüpft die beiden zuletzt eingegebenen Zahlen miteinander.

Folgende Eingaben in den Rechner sind möglich :

(**RET** heißt hier : Return- oder Enter-Taste drücken)

1) Komplexe Zahlen

Komplexe Zahlen können dem Programm übergeben werden durch

a) Eingabe von Real- und Imaginärteil

Beispiel z = 5 - 6i

Eingabe : 5 **RET** -6 **RET**

Wurzeln können direkt eingegeben werden :

z.B. $\sqrt{2}$ ist einzugeben als **w2** , $-\sqrt{5}$ als **-w5**.

Beispiel $z = \sqrt{3} - \sqrt{2}i$

Eingabe : w3 **RET** -w2 **RET**

b) Eingabe von Betrag und Winkel in Grad (Polarform)

Diese Eingabe ist durch Betätigen der Taste '**c**' einzuleiten. z wird cartesisch angezeigt und gespeichert.

Beispiel : $z = 4 \cdot e^{i\frac{\pi}{6}}$ ($\frac{\pi}{6} \hat{=} 30°$)

Eingabe : c **RET** 4,30 **RET**

Ausgabe : 3.46410162 + 2i (ist im Register abgelegt)

---

*) gegebenenfalls PI = 4*ATN(1) definieren !

2) Rechenoperationen + , - , * , / , ↑k (k ganzzahlig)

Beispiel : $z = \dfrac{(1 + 2i)(3 - 4i)}{1 - i}$

Eingabe : 1 **RET** 2 **RET** 3 **RET** -4 **RET** * **RET** 1 **RET** -1 **RET** / **RET**

Als Zwischenergebnis erhält man den Zähler 11+2i und dann 4.5 + 6.5·i als Ergebnis.

Beispiel : $z = \dfrac{(1 + \sqrt{3}i)^2 (1 - i)^2}{(-\sqrt{3} + i)^3}$

Eingabe : 1 **RET** w3 **RET** ↑2 **RET** 1 **RET** -1 **RET** ↑2 **RET** * **RET**
-w3 **RET** 1 **RET** ↑3 **RET** / **RET**

Ergebnis : 0.5 - 0.866025404·i

3) Rechenanweisungen ↑1/k , p

a) ↑1/k (k-te Wurzeln) : Es ergeben sich die k Lösungen der Gleichung $z^k = w$, wobei w die zuvor eingegebene Zahl ist.

Beispiel $z^3 = -8i$

Eingabe : 0 **RET** -8 **RET** ↑1/3 **RET**

Ausgabe : 0-te Lösung : z = 1.73205081 - i
Betrag : 2 , Winkel : -30°
1-te Lösung : z = 2i
Betrag : 2 , Winkel : 90°
2-te Lösung : z = -1.73205081 - i
Betrag : 2 , Winkel : 210°

b) p (Polarform) : ergibt die Polardarstellung der zuvor eingegebenen komplexen Zahl.

Beispiel : $z = \sqrt{2} + \sqrt{6}\cdot i$ in Polarform

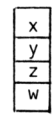
Eingabe : w2 **RET** w6 **RET** p **RET**

Ausgabe : Betrag 2.82842713 Winkel 60° $\hat{=} \frac{\pi}{3}$.

Anmerkung : Die Ergebnisse von Rechenoperationen werden vom Programm zur Weiterverarbeitung im Stapelregister abgelegt. Die Ergebnisse der Rechenanweisungen jedoch werden nur angezeigt, sie verändern nicht die Stapelregister.

4) andere Anweisungen

a : Anzeige der Inhalte der 4 Register des Stapels :



x : Löschen des x-Registers

v : Vertauschen von x- und y-Register

sn : Speichern des x-Registerinhalts auf dem Speicherplatz n (n ganz, $0 \leq n \leq 9$)

rn : Rückruf des Inhalts von Speicherplatz n in das x-Register.

```
10 CLS:PRINT"Rechnen mit komplexen Zahlen":PRINT
20 DIM z(3,1):DIM s(9,1)
30 INPUT ;o$
40 IF o$="+" THEN 280
50 IF o$="-" THEN 300
60 IF o$="*" THEN 320
70 IF o$="/" THEN 350
80 IF LEFT$(o$,1)="↑" THEN 400
90 IF o$="a" THEN GOSUB 670:GOTO 30
100 IF o$="c" THEN GOTO 770
110 IF o$="v" THEN GOSUB 700:GOTO 30
120 IF o$="p" THEN GOSUB 730:GOTO 30
130 IF o$="x" THEN GOSUB 600:GOTO 30
140 IF o$="e" THEN END
150 IF LEFT$(o$,1)="s" THEN GOSUB 850:GOTO 30
160 IF LEFT$(o$,1)="r" THEN GOSUB 870:GOTO 30
170 GOSUB 190:x=t
180 INPUT o$:GOSUB 190:y=t:GOSUB 240:GOTO 30
190 IF LEFT$(o$,2)="-w" THEN 220
200 IF LEFT$(o$,1)<>"w" THEN t=VAL(o$):RETURN
210 t=SQR(VAL(MID$(o$,2,LEN(o$)-1))):RETURN
220 t=-SQR(VAL(MID$(o$,3,LEN(o$)-2))):RETURN
230 ´Ablage in das 1.Register
240 FOR i=3 TO 1 STEP -1
250 z(i,0)=z(i-1,0):z(i,1)=z(i-1,1):NEXT
260 z(0,0)=x:z(0,1)=y:RETURN
```

```
270 ' +
280 FOR i=0 TO 1:z(0,i)=z(0,i)+z(1,i):NEXT:GOTO 550
290 ' -
300 FOR i=0 TO 1:z(0,i)=-z(0,i)+z(1,i):NEXT:GOTO 550
310 ' *
320 x=z(0,0):y=z(0,1):z(0,0)=x*z(1,0)-y*z(1,1)
330 z(0,1)=z(1,0)*y+z(1,1)*x:GOTO 550
340 ' /
350 x=z(0,0):y=z(0,1):n=z(0,0)*z(0,0)+z(0,1)*z(0,1)
360 z(0,0)=(x*z(1,0)+y*z(1,1))/n
370 z(0,1)=(-z(1,0)*y+z(1,1)*x)/n
380 GOTO 550
390 ' ↑
400 IF MID$(o$,3,1)="/" THEN 460
410 ' ↑k
420 n=VAL(RIGHT$(o$,LEN(o$)-1))
430 GOSUB 810:r=r↑n:w=w*n
440 z(0,0)=r*COS(w):z(0,1)=r*SIN(w):GOTO 570
450 ' ↑1/k
460 PRINT:n=VAL(RIGHT$(o$,LEN(o$)-3))
470 GOSUB 810:r=r↑(1/n):w=w/n
480 FOR k=0 TO n-1
490 v=w+k*2*PI/n:x=r*COS(v):GOSUB 630
500 PRINT k"te Lsg : Re z="x;
510 x=r*SIN(v):GOSUB 630:PRINT"  Im z="x
520 PRINT"            Betrag"r" Winkel :"v*180/PI
530 NEXT :GOTO 30
540 ' Herunterziehen von 3.und 4.Register
550 FOR j=1 TO 2:FOR i=0 TO 1:z(j,i)=z(j+1,i):NEXT:NEXT
560 ' Anzeige 1.Register
570 PRINT:x=z(0,0):GOSUB 630:PRINT"Re z=";x;
580 x=z(0,1):GOSUB 630:PRINT"Im z=";x:GOTO 30
590 ' Herunterziehen von 2.,3.und 4.Register
600 PRINT:FOR j=0 TO 2:FOR i=0 TO 1
610 z(j,i)=z(j+1,i):NEXT:NEXT:RETURN
620 ' Rundung
630 x=100*x:IF(x-INT(x)<0.001) THEN x=INT(x)/100:RETURN
```

```
640 IF(INT(x)-x+1<0.001) THEN x=(INT(x)+1)/100:RETURN
650 x=x/100:RETURN
660 ' Registeranzeige
670 PRINT:FOR i=1 TO 4
680 PRINT i". Reg :";z(i-1,0);z(i-1,1):NEXT :RETURN
690 ' 1.un 2. Register vertauschen
700 PRINT:x=z(0,0):y=z(0,1):z(0,0)=z(1,0)
710 z(0,1)=z(1,1):z(1,0)=x:z(1,1)=y:RETURN
720 ' Ausgabe des 1.Registers in Polarform"
730 PRINT:GOSUB 810
740 PRINT"Betrag :"r"  Winkel :";
750 PRINT w*180/PI"Grad  =  "w/PI"*pi Rad":GOTO 30
760 ' Eingabe in Polarform
770 PRINT:PRINT"Eingabe :Betrag,Winkel in Grad"
780 INPUT;r,w:w=w*PI/180
790 x=r*COS(w):y=r*SIN(w):GOSUB 240:GOTO 570
800 ' 1.Register in Polarform
810 x=z(0,0):y=z(0,1):r=SQR(x*x+y*y)
820 IF x=0 THEN w=PI*SGN(y)/2:RETURN
830 w=ATN(y/x)+PI*(1-SGN(x))/2:RETURN
840 ' Speichern
850 s=VAL(RIGHT$(o$,1)):s(s,0)=z(0,0):s(s,1)=z(0,1):RETURN
860 ' Rueckruf aus Speicher
870 s=VAL(RIGHT$(o$,1)):x=s(s,0):y=s(s,1)
880 GOSUB 240:GOSUB 570:RETURN
```

Beispiel : Berechnen Sie die Lösungen der Gleichung

$$z^4 = 4 \cdot e^{2\pi/3} \cdot (2 + 2\sqrt{3} \cdot i).$$

Da $2\sqrt{3}$ nicht direkt eingegeben werden kann, wird umgeformt

$$z^4 = 8 \cdot e^{2\pi/3} \ (1 + \sqrt{3} \cdot i).$$

Eingabe : c **RET** 8,120 **RET** 1 **RET** w3 **RET** * **RET** ↑1/4 **RET**

Zwischenergebnis : $z^4 = -16$

Ergebnis : $-\sqrt{2} - \sqrt{2}i$, $-\sqrt{2} + \sqrt{2}i$, $\sqrt{2} - \sqrt{2}i$, $\sqrt{2} + \sqrt{2}i$

Beispiel : Bestimmen Sie Real- und Imaginärteil von

$$z = 4 \cdot e^{\pi/3} - 5 \cdot e^{\pi/12} .$$

Eingabe : c **RET** 4,60 **RET** c **RET** 5,15 **RET** - **RET**

Ergebnis : -2.82962913 + 2.17·i.

Quadratische Gleichung (komplex)*

Es werden die Lösungen der quadratischen Gleichung

$$a_2 z^2 + a_1 z + a_0 = 0 \quad , \quad a_2 \neq 0$$

bestimmt. Die komplexen Koeffizienten a_k sind dazu in der Form Re z,Im z einzugeben.

```
10 CLS:PRINT"Quadratische Gleichung mit"
20 PRINT"komplexen Koeffizienten":PRINT
30 '
40 PRINT"Koeffizienteneingabe : Re,Im":PRINT
50 FOR k=2 TO 0 STEP-1
60 PRINT"a("k") :";:INPUT a(k),b(k)
70 NEXT k
80 '
90 x=a(1)*a(1)-b(1)*b(1)-4*(a(0)*a(2)-b(0)*b(2))
100 y=2*a(1)*b(1)-4*(b(0)*a(2)+a(0)*b(2))
110 n=2*(a(2)*a(2)+b(2)*b(2))
120 IF x=0 THEN w=PI*SGN(y)/4:GOTO 140
130 w=ATN(y/x)/2+PI*(1-SGN(x))/4
140 r=(x*x+y*y):IF r<10↑(-7) THEN 230
150 r=r↑(1/4)
160 x=r*COS(w):y=r*SIN(w):PRINT
170 u=-a(1)-x:v=-b(1)-y:GOSUB 220:PRINT"z1=";:GOSUB 210
180 u=-a(1)+x:v=-b(1)+y:GOSUB 220:PRINT"z2=";:GOSUB 210
190 GOTO 250
200 '
210 PRINT p/n"+i*"q/n:RETURN
220 p=u*a(2)+v*b(2):q=v*a(2)-u*b(2):RETURN
230 p=-a(1)*a(2)-b(1)*b(2):q=a(1)*b(2)-a(2)*b(1)
240 PRINT:PRINT"z1/2=";:GOSUB 210
250 END
```

Beispiel 1 $\quad iz^2 - (1+3i)z + 2+2i = 0$

$$z_1 = 2 \qquad z_2 = 1-i$$

Beispiel 2 $\quad (3-i)z^2 - (18+14i)z - 3+41i = 0$

$$z_{1,2} = 2+3i$$

*) gegebenenfalls PI = 4*ATN(1) definieren!

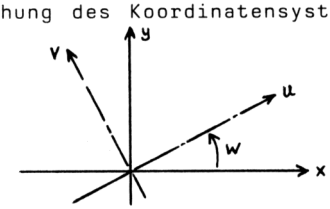
Kegelschnitte*

Dieses Programm bestimmt die Lösungsmenge der Gleichung :

(*) $ax^2 + bxy + cy^2 + dx + ey + f = 0$.

Abgesehen von den Sonderfällen (leere Lösungsmenge bzw. jedes (x,y) gehört zur Lösung) beschreibt diese Gleichung einen Kegelschnitt.Die Lösung kann dann sein :
Genau ein Punkt, eine oder zwei Geraden, eine Ellipse (speziell ein Kreis), eine Parabel oder eine Hyperbel.

Der Fall b=0 läßt sich einfach durchschauen und tatsächlich läßt sich durch Drehung des Koordinatensystems dieser Fall stets herleiten:



Durch Drehung um den Winkel w erhält man das u,v-System. Als Zusammenhang zwischen den alten und den neuen Koordinaten kann man herleiten :

$x = u \cdot \cos w - v \cdot \sin w$ $\qquad$ $u = x \cdot \cos w + y \cdot \sin w$

$y = u \cdot \sin w + v \cdot \cos w$ $\qquad$ $v = -x \cdot \sin w + y \cdot \cos w$

Bei geeigneter Wahl von w geht (*) über in

(**) $\bar{a}u^2 + \bar{c}v^2 + \bar{d}u + \bar{e}v + \bar{f} = 0$

und dieser Übergang heißt Hauptachsentransformation.

Nach Eingabe der Koeffizienten a,b,c,d,e und f führt das Programm auf einen der folgenden Fälle :

Fall 0 : Die Lösungsmenge ist leer

Fall 1 : Ein Punkt P_0
im x,y-System : $P_0=(P,Q)$
im u,v-System : $P_0=(R,S)$

Fall 2 : Eine Gerade g
im x,y-System g : $ax + by = c$
im u,v System g : $du + ev = c$

Fall 3 : Ein paralleles Geradenpaar $g_{1,2}$
im x,y-System $g_{1,2}$: $ax + by = c_{1,2}$
im u,v-System $g_{1,2}$: $du + ev = c_{1,2}$

*) gegebenenfalls PI = 4*ATN(1) definieren!

Fall 4 : Ein nicht-paralleles Geradenpaar

Schnittpunkt P_0

im x,y-System $g_{1,2} : a_{1,2}(y-Q)=b_{1,2}(x-P)$ $P_0=(P,Q)$

im u,v-System $g_{1,2} : a_{1,2}(v-S)=b_{1,2}(u-R)$ $P_0=(R,S)$

Fall 5 : Eine Ellipse mit Mittelpunkt P_0 und Halbmessern a,b

im u,v-System $\frac{(u-R)^2}{a^2} + \frac{(v-S)^2}{b^2} = 1$, $P_0=(R,S)$

Mittelpunkt im x,y-System $P_0=(P,Q)$

Fall 6 : Ein Kreis um P_0 mit Radius a

im x,y-System $(x-P)^2 + (y-Q)^2 = a^2$, $P_0=(P,Q)$

Fall 7 : Eine Hyperbel mit Mittelpunkt P_0 und Parametern a,b

im u,v-System $\frac{(u-R)^2}{a^2} - \frac{(v-S)^2}{b^2} = 1$, $P_0=(R,S)$

Mittelpunkt im x,y-System $P_0=(P,Q)$.

Die Hyperbel ist in u-Richtung geöffnet.

Fall 8 : Eine Hyperbel mit Mittelpunkt P_0 und Parametern a,b

im u,v-System $-\frac{(u-R)^2}{a^2} + \frac{(v-S)^2}{b^2} = 1$, $P_0=(R,S)$

Mittelpunkt im x,y-System $P_0=(P,Q)$

Die Hyperbel ist in v-Richtung geöffnet.

Fall 9 : Eine Parabel mit Scheitel P_0 und Parameter a

im u,v-System $(v-S)^2 = a(u-R)$, $P_0=(R,S)$

Scheitel im x,y-System $P_0=(P,Q)$

Für $a>0$ ist die Parabel in +u-Richtung, für $a<0$ in -u-Richtung geöffnet.

Fall 10: Eine Parabel mit Scheitel P_0 und Parameter a

im u,v-System $(u-R)^2 = a(v-S)$, $P_0=(R,S)$

Scheitel im x,y-System $P_0=(P,Q)$

Für $a>0$ ist die Parabel in +v-Richtung, für $a<0$ in -v-Richtung geöffnet.

Fall 11: Jeder Punkt (x,y) ist Lösung.

1 Kegelschnitte

Das nachfolgende Programm macht in den einzelnen Fällen die zugehörigen Angaben. In der dann folgenden verkürzten Version endet das Programm mit z.B. 'Fall 7'. Die zugehörigen Angaben :Drehwinkel w,Mittelpunkt P_0 und Parameter a,b sind dann durch Abfrage zu ermitteln.

```
10 CLS:GOSUB 730:' Ueberschrift
20 '  Koeffizienteneingabe
30 INPUT"a=";a:INPUT"b=";b:INPUT"c=";c
40 INPUT"d=";d:INPUT"e=";e:INPUT"f=";f
50 w=0:x=0:t=1:u=0
60 IF b=0 THEN 160
70 ' Drehwinkel
80 w=PI/4:IF a<>c THEN w=ATN(b/(a-c))/2
90 t=COS(w):u=SIN(w)
100 ' neue Koeffizienten
110 x=a*t*t+b*t*u+c*u*u:GOSUB 560:k=x
120 x=a*u*u-b*t*u+c*t*t:GOSUB 560:l=x
130 x=d*t+e*u:GOSUB 560:m=x
140 x=e*t-d*u:GOSUB 560:n=x
150 a=k:c=l:d=m:e=n:f1=f
160 IF a<>0 THEN 340
170 IF c<>0 THEN 220
180 IF d<>0 OR e<>0 THEN 210
190 IF f<>0 THEN z=0:GOTO 610
200 z=11:GOTO 610
210 a=d*t-e*u:b=d*u+e*t:c=-f:z=2:GOTO 600
220 x=(e*e-4*c*f)/4/c/c:GOSUB 560:q=x
230 s=-e/2/c
240 IF d<>0 THEN 320
250 a=-u:b=t:d=0:e=1
260 IF q<0 THEN z=0:GOTO 610
270 IF q=0 THEN 300
280 c1=s-SQR(q):c2=s+SQR(q):z=3:IF xx=1 THEN 360
290 GOTO 600
300 c=s:z=2:IF xx=1 THEN 360
```

```
310 GOTO 600
320 a=-d/c:r=c*q/d:IF xx=1 THEN 370
330 z=9:GOTO 600
340 IF c<>0 THEN 380
350 c=a:g=d:d=e:e=g:xx=1:GOTO 220
360 a=t:b=u:e=0:d=1:GOTO 600
370 g=r:r=s:s=g:z=10:GOTO 600
380 r=-d/2/a:s=-e/2/c
390 x=d*d/4/a+e*e/4/c-f:GOSUB 560:q=x
400 IF q<>0 THEN 480
410 x=-a/c:GOSUB 560:a=x
420 IF a<0 THEN z=1:GOTO 600
430 IF a=0 THEN a=-u:b=t:c=s:z=2:GOTO 600
440 m=SQR(a):v=ATN(m)
450 x=COS(w+v):GOSUB 560:a1=x:x=SIN(w+v):GOSUB 560:b1=x
460 x=COS(w-v):GOSUB 560:a2=x:x=SIN(w-v):GOSUB 560:b2=x
470 z=4:GOTO 600
480 a=a/q:c=c/q:IF a<0 THEN 530
490 IF c<0 THEN a=1/SQR(a):b=1/SQR(-c):z=7:GOTO 600
500 a=1/SQR(a):b=1/SQR(c)
510 IF ABS(a-b)<10↑(-6) THEN z=6:GOTO 600
520 z=5:GOTO 600
530 IF c<0 THEN z=0:GOTO 610
540 a=1/SQR(-a):b=1/SQR(c):z=8:GOTO 600
550 ' Runden
560 x=1000*x:IF x-INT(x)<10↑(-2) THEN x=INT(x):GOTO 580
570 IF INT(x)+1-x<10↑(-2) THEN x=INT(x)+1
580 x=x/1000:RETURN
590 '
600 x=r*t-s*u:GOSUB 560:p=x:x=r*u+s*t:GOSUB 560:q=x
610 PRINT:PRINT"Fall "z
620 '
630 ' Textausgabe
640 c$=" im x,y-Syst."
650 a$="("+STR$(p)+"/"+STR$(q)+")"+c$
660 b$="("+STR$(r)+"/"+STR$(s)+") im u,v-Syst."
670 IF (w=0 OR z<5) THEN 720
```

```
680 PRINT:PRINT"Drehwinkel"w*180/PI" Grad"
690 PRINT"Im gedrehten System lautet die Gleichung"
700 PRINT k"*u↑2+"l"*v↑2+"m"*u+"n"*v+"f1"=0":PRINT
710 '
720 ON z+1 GOTO 770,790,810,860,920,1010,1060,1100,
    1100,1180,1180,1250
730 PRINT"Es wird die Loesungsmenge der Gleichung":PRINT
740 PRINT"         ax^2+bxy+cy^2+dx+ey+f=0":PRINT
750 PRINT"bestimmt":PRINT:RETURN
760 '
770 PRINT"Die Loesungsmenge ist leer":END
780 '
790 PRINT"Ein Punkt :"a$:END
800 '
810 PRINT"Eine Gerade :":PRINT
820 IF b=0 THEN PRINT"x="c/a:END
830 IF a<>0 THEN PRINT a/b"*x+y="c/b:END
840 PRINT"y="c/b:END
850 '
860 PRINT"Ein paralleles Geradenpaar :":PRINT
870 IF b=0 THEN PRINT "x="c1/a:PRINT "x="c2/a:END
880 IF a=0 THEN PRINT"y="c1/b:PRINT"y="c2/b:END
890 PRINT a/b"*x+y="c1/b
900 PRINT a/b"*x+y="c2/b:PRINT:END
910 '
920 PRINT"Ein nicht paralleles Geradenpaar :":PRINT
930 IF a1=0 THEN PRINT"x="p:GOTO 960
940 IF b1<>0 THEN PRINT "y-"q"="b1/a1"*(x-"p")":GOTO 960
950 PRINT"y="q
960 IF a2=0 THEN PRINT"x="p:GOTO 990
970 IF b2<>0 THEN PRINT "y-"q"="b2/a2"*(x-"p")":GOTO 990
980 PRINT"y="q
990 PRINT:PRINT"Schnittpunkt "+a$:END
1000 '
1010 PRINT"Eine Ellipse mit Mittelpunkt M":PRINT
1020 PRINT"M="a$:IF w=0 THEN 1040
1030 PRINT b$:PRINT
1040 PRINT"a↑2=";a*a;" b↑2=";b*b:END
```

```
1050 '
1060 PRINT"Ein Kreis":PRINT
1070 PRINT"Mittelpunkt :"a$
1080 PRINT"Radius :"a:END
1090 '
1100 PRINT"Eine Hyperbel mit Mittelpunkt M":PRINT
1110 PRINT"M="a$:IF w=0 THEN b$=""
1120 PRINT b$:PRINT"a↑2=";a*a;" b↑2=";b*b:PRINT
1130 IF w=0 THEN r$="x":IF z=8 THEN r$="y"
1140 IF w<>0 THEN r$="u":IF z=8 THEN r$="v"
1150 PRINT"Die Hyperbel ist in "r$"-Richtung geoeffnet":END
1160 '
1170 PRINT"Eine Parabel mit Scheitel S":PRINT
1180 PRINT"S="a$:IF w=0 THEN b$=""
1190 PRINT b$;:PRINT
1200 IF w=0 THEN r$="x":IF z=10 THEN r$="y"
1210 IF w<>0 THEN r$="u":IF z=10 THEN r$="v"
1220 PRINT:v$="-":IF a>0 THEN v$="+"
1230 PRINT"Die Parabel ist in "v$r$" Richtung geoeffnet":END
1240 '
1250 PRINT"Jedes (x,y) genuegt der Gleichung":END
```

Beispiel 0 : $2x^2 - 2xy + y^2 - 4y + 9 = 0$

Die Lösungsmenge ist leer.

Beispiel 1 : $3x^2 - 4xy + 4y^2 - 8x + 8 = 0$

Ein Punkt : (2/1)

Beispiel 2 : $4x^2 - 12xy + 9y^2 + 4x - 6y + 1 = 0$

Eine Gerade : $-2x + 3y = 1$ (nach angemessener Umformung)

Beispiel 3 : $4x^2 + 12xy + 9y^2 + 4x + 6y - 8 = 0$

Zwei parallele Geraden : $2x + 3y = -4$, $2x + 3y = 2$
(nach Umformung)

Beispiel 4 : $3x^2 - 4xy + y^2 + 26x - 16y + 55 = 0$

Zwei nicht parallele Geraden , Schnittpunkt (-3/2)
$y - 2 = 1(x + 3)$, $y - 2 = 3(x + 3)$

Beispiel 5 : $x^2 + 4xy + 5y^2 - 6x - 8y + 12 = 0$

Eine Ellipse mit Mittelpunkt (7/-2) im x,y-System. Die Ellipse ist achsenparallel zum um -22,5° gedrehten u,v-System und wird in diesem System (nach Rundung) beschrieben durch $0{,}17u^2 + 5{,}83v^2 - 2{,}48u - 9{,}69v + 12 = 0$. Für die Halbmesser a,b gilt (gerundet) $a^2=5{,}89$ $b^2=0{,}17$.

Beispiel 6 : $x^2 + y^2 + 4x - 6y + 4 = 0$

Ein Kreis um (-2/3) mit Radius 3.

Beispiel 7 : $3x^2 - 10xy + 3y^2 + 14x - 2y + 3 = 0$

Eine Hyperbel mit Mittelpunkt (1/2) im x,y-System. Die Hyperbel ist achsenparallel zum um 45° gedrehten u,v-System und wird in diesem System (nach Rundung) beschrieben durch $-2u^2 + 8v^2 + 8{,}49u - 11{,}31v + 3 = 0$. Für die Parameter a,b gilt $a^2=4$ $b^2=1$. Öffnung in u-Richtung.

Beispiel 8 : $xy - 3x + 2y - 5 = 0$

Eine Hyperbel mit Mittelpunkt (-2/3) im x,y-System. Die Hyperbel ist achsenparallel zum um 45° gedrehten u,v-System und wird in diesem System (nach Rundung) beschrieben durch $0{,}5u^2 - 0{,}5v^2 - 0{,}71u + 3{,}54v - 5 = 0$. Für die Parameter a,b gilt $a^2 = b^2 = 2$. Öffnung in v-Richtung.

Beispiel 9 : $y^2 + 4x - 2y + 17 = 0$

Eine Parabel mit Scheitel in (-4/1) im x,y-System. Die Parabel ist im x,y-System achsenparallel und sie ist in -x-Richtung geöffnet.

Beispiel 10 : $x^2 + 2xy + y^2 + 7x + 5y + 8 = 0$

Eine Parabel mit Scheitel (-1/-2) im x,y-System Die Parabel ist achsenparallel zum um 45° gedrehten u,v-System und wird in diesem System (nach Rundung) beschrieben durch $2u^2 + 8{,}48u - 1{,}41v + 8 = 0$. Die Parabel ist in +v-Richtung geöffnet.

2 Kurzprogramm *

Das Programm gibt an, welcher Lösungsfall vorliegt,und die zugehörigen Größen (s.vorangehende Liste der Fälle) können abgefragt werden.

```
10 INPUT"a=";a:INPUT"b=";b:INPUT"c=";c
20 INPUT"d=";d:INPUT"e=";e:INPUT"f=";f
30 w=0:x=0:t=1:u=0:xx=0
40 IF b=0 THEN 130
50 w=PI/4:IF a<>c THEN LET w=ATN(b/(a-c))/2
60 t=COS(w):u=SIN(w)
70 x=a*t*t+b*t*u+c*u*u:GOSUB 540:k=x
80 x=a*u*u-b*t*u+c*t*t:GOSUB 540:l=x
90 x=d*t+e*u:GOSUB 540:m=x
100 x=e*t-d*u:GOSUB 540:n=x
110 a=k:c=l:d=m:e=n
120 k=a:l=c:m=d:n=e:f1=f
130 IF a<>0 THEN 310
140 IF c<>0 THEN 190
150 IF (d<>0 OR e<>0) THEN 180
160 IF f<>0 THEN LET z=0:GOTO 530
170 z=11:GOTO 530
180 a=d*t-e*u:b=d*u+e*t:c=-f:z=2:GOTO 520
190 x=(e*e-4*c*f)/4/c/c:GOSUB 540:q=x
200 s=-e/2/c
210 IF d<>0 THEN 290
220 a=-u:b=t:d=0:e=1
230 IF q<0 THEN LET z=0:GOTO 530
240 IF q=0 THEN 270
250 c1=s-SQR(q):c2=s+SQR(q):z=3:IF xx=1 THEN 330
260 GOTO 520
270 c=s:z=2:IF xx=1 THEN 330
280 GOTO 520
290 a=-d/c:r=c*q/d:IF xx=1 THEN 340
300 z=9:GOTO 520
310 IF c<>0 THEN 350
320 c=a:g=d:d=e:e=g:xx=1:GOTO 190
330 a=t:b=u:e=0:d=1:GOTO 520
```

*) gegebenenfalls PI = 4*ATN(1) definieren!

```
340 g=r:r=s:s=g:z=10:GOTO 520
350 r=-d/2/a:s=-e/2/c
360 x=d*d/4/a+e*e/4/c-f:GOSUB 540:q=x
370 IF q<>0 THEN 450
380 x=-a/c:GOSUB 540:a=x
390 IF a<0 THEN LET z=1:GOTO 520
400 IF a=0 THEN LET a=-u:b=t:c=s:z=2:GOTO 520
410 m=SQR(a):v=ATN(m)
420 a1=COS(w+v):b1=SIN(w+v)
430 a2=COS(w-v):b2=SIN(w-v)
440 z=4:GOTO 520
450 a=a/q:c=c/q:IF a<0 THEN 500
460 IF c<0 THEN LET a=1/SQR(a):b=1/SQR(-c):z=7:GOTO 520
470 a=1/SQR(a):b=1/SQR(c)
480 x=a-b:GOSUB 540:IF x=0 THEN LET z=6:GOTO 520
490 z=5:GOTO 520
500 IF c<0 THEN z=0:GOTO 530
510 a=1/SQR(-a):b=1/SQR(c):z=8
520 p=r*t-s*u:q=r*u+s*t
530 PRINT:PRINT"Fall "z:END
540 IF ABS(x)<10↑(-6) THEN LET x=0
550 RETURN
```

Beispiel 1 $x^2 + 2xy - 3y^2 - 4x - 8y + 3 = 0$

Das Programm liefert : Fall 4

Durch Abfrage erhält man (gerundet) :

a1=0,707 b1=0,707 a2=0.948 b2=-0,316 P=2,5 Q=-0,5

Da a2/b2=-3 , erhält man die beiden Geraden

$-3(y + 0,5)=x - 2,5$ und $y + 0,5 = x - 2,5$

mit dem Schnittpunkt (2,5/-0,5).

Beispiel 2 $x^2 - 2xy + y^2 - 7x + 5y + 8 = 0$

Das Programm ermittelt : Fall 9.

Aus der Liste der Lösungsfälle entnimmt man, daß Drehwinkel w, Scheitel (P,Q) und Parameter a zu bestimmen sind.

w*180/PI=45 P=1 Q=-2 a=0.707 Öffnung in +u-Richtung.

Bruchrechnung

Dieses Programm erlaubt eine 'exakte' Bruchrechnung. Brüche werden also tatsächlich als Brüche und nicht als Dezimalzahlen verarbeitet.
Die Arithmetik arbeitet nach der 'umgekehrten polnischen Notation' : Eingegebene Brüche werden auf einem 'Stapel' abgelegt und eine nachfolgende Rechenoperation verknüpft die beiden zuletzt eingegebenen Brüche miteinander.

Beispiel : $\frac{1}{308} + \frac{2}{917}$

Eingabe : 1 **RET** 308 **RET** 2 **RET** 917 **RET** + **RET**

RET heißt hier : RETURN oder ENTER-Taste drücken.

Ergebnis : 219/40348

Folgende Eingaben in den Rechner sind möglich :

1) Brüche aus ganzen Zahlen
   Beispiel : -7 **RET** -21 **RET**
2) Rechenoperationen + , - , * , / , ↑k (k ganzzahlig)
   Beispiel $\frac{2}{3}(\frac{5}{6} + \frac{4}{9})$
   Eingabe : 2 **RET** 3 **RET** 5 **RET** 6 **RET** 4 **RET** 9 **RET** + **RET** * **RET**
   Ergebnis : Nach + **RET** erscheint die Summe 23/18
   nach * **RET** das Ergebnis 23/27.
   Beispiel $((\frac{2}{3})^{-2} + \frac{1}{2})\ \frac{4}{3}$
   Eingabe : 2 **RET** 3 **RET** ↑ -2 **RET** 1 **RET** 2 **RET** + **RET** 4 **RET** 3 **RET** * **RET**
   Ergebnis : Als Zwischenergebnisse erhält man 9/4 , dann 11/4 und schließlich 11/3.
3) andere Anweisungen : a , x , e , k
   a : Anzeige der 4 Register des Stapels
   x : Löschen der letzten Registereintragung
   e : Ende des Programms
   k : Kürzen des zuletzt eingegebenen Bruchs

```
10 CLS:PRINT"Bruchrechnung":PRINT
20 DIM z(3,1)
30 INPUT;o$
40 IF o$="+" THEN 190
50 IF o$="-" THEN 220
60 IF o$="*" THEN 250
70 IF o$="/" THEN 270
80 IF LEFT$(o$,1)="↑" THEN 300
90 IF o$="a" THEN 370
100 IF o$="k" THEN 340
110 IF o$="x" THEN GOTO 520
120 IF o$="e" THEN END
130 x=VAL(o$):INPUT "/",y
140 IF y=0 THEN GOSUB 350:GOTO 30
150 ´ Ablage in das 1.Register
160 FOR i=3 TO 1 STEP -1:z(i,0)=z(i-1,0):z(i,1)=z(i-1,1):NEXT
170 z(0,0)=x:z(0,1)=y:GOTO 30
180 ´ +
190 a=z(0,0)*z(1,1)+z(1,0)*z(0,1):b=z(0,1)*z(1,1)
200 GOSUB 470:GOTO 400
210 ´ -
220 a=-z(0,0)*z(1,1)+z(1,0)*z(0,1):b=z(0,1)*z(1,1)
230 GOSUB 470:GOTO 400
240 ´ *
250 a=z(0,0)*z(1,0):b=z(0,1)*z(1,1):GOSUB 470:GOTO 400
260 ´ /
270 IF z(0,0)=0 THEN GOSUB 350:GOTO 520
280 b=z(0,0)*z(1,1):a=z(0,1)*z(1,0):GOSUB 470:GOTO 400
290 ´ ↑k
300 k=VAL(RIGHT$(o$,LEN(o$)-1)):IF k<0 THEN k=-k:GOTO 320
310 PRINT:a=z(0,0)↑k:b=z(0,1)↑k:GOSUB 470:GOTO 410
320 PRINT:b=z(0,0)↑k:a=z(0,1)↑k:GOSUB 470:GOTO 410
330 ´ kuerzen
340 a=z(0,0):b=z(0,1):GOSUB 470:PRINT:GOTO 410
350 PRINT" keine Division durch 0":RETURN
360 ´ Anzeige aller Register
370 PRINT:FOR k=0 TO 3
```

```
380 PRINT k+1"tes Reg.:"z(k,0)"/"z(k,1):NEXT :GOTO 30
390 ' Herunterziehen von 3.und 4.Register
400 PRINT:FOR j=1 TO 2:FOR i=0 TO 1:z(j,i)=z(j+1,i):NEXT:NEXT
410 z(0,0)=z/g:z(0,1)=n/g
420 ' Anzeige 1.Register
430 IF z(0,0)=0 THEN PRINT"=0":GOTO 30
440 IF z(0,1)=1 THEN PRINT"="z(0,0):GOTO 30
450 PRINT"=";z(0,0);:PRINT"/";z(0,1):GOTO 30
460 ' ggT bestimmen
470 z=a:n=b
480 g=b:b=a-INT(a/b)*b:a=g
490 IF b=0 THEN RETURN
500 GOTO 480
510 ' Herunterziehen von 2.,3.und 4.Register
520 PRINT:FOR j=0 TO 2:FOR i=0 TO 1:z(j,i)=z(j+1,i):NEXT:NEXT:
    GOTO 30
```

Beispiel : $\dfrac{(\frac{1}{2} + \frac{2}{3})\cdot(\frac{4}{5} + \frac{6}{7})}{(\frac{5}{8} + \frac{4}{3})\cdot(3 + \frac{3}{7})}$

Eingabe : 1 **RET** 2 **RET** 2 **RET** 3 **RET** + **RET** 4 **RET** 5 **RET** 6 **RET**
7 **RET** + **RET** * **RET** das war der Zähler
5 **RET** 8 **RET** 4 **RET** 3 **RET** + **RET** 3 **RET** 1 **RET** 3 **RET**
7 **RET** + **RET** * **RET** das war der Nenner
/ **RET**

Ergebnis : 203/705 .

Beispiel : Der Bruch $\frac{1921}{4633}$ ist zu kürzen.

Eingabe : 1921 **RET** 4633 **RET** k **RET**

Ergebnis : 17/41

Größter gemeinsamer Teiler

Das Programm bestimmt den GGT (a,b) zweier ganzer Zahlen a,b mit dem Euklidschen Divisionsalgorithmus.
Der GGT ist darstellbar als Vielfachsumme von a und b :

$$(a,b) = x \cdot a + y \cdot b \quad \text{mit } x,y \in \mathbb{Z}$$

Das Programm ermittelt alle ganzzahligen Lösungen x,y.

```
10 CLS:PRINT"Groesster gemeinsamer Teiler zweier"
20 PRINT"ganzer Zahlen a,b":PRINT
30 INPUT "a=";a:INPUT "b=";b
40 IF (INT(a)<a OR INT(b)<b) THEN 30
50 d=a:e=b:o=1:l=1
60 g=b:f=INT(a/b):b=a-f*b:a=g
70 x=l:y=p:l=k-f*l:p=o-f*p:k=x:o=y
80 IF b=0 THEN 100
90 GOTO 60
100 PRINT:PRINT ABS(g)" ist ggT":PRINT
110 PRINT g"=("k"+k*"d/g")*"e"+("o"-k*"e/g")*"d
120 PRINT"    k beliebig ganzzahlig"
```

Beispiel 1 a=17 b=30

1 ist GGT und

1 = x*30 + y*17 mit x=4+17*k , y=-7-30*k,

wobei k eine beliebige ganze Zahl bedeutet.

Beispiel 2 Bestimmen Sie alle ganzzahligen Lösungen der Gleichung 15x + 21y = 33.

Man ermittelt zunächst den GGT von 15 und 21.Dies ist 3.

3 = (-2+5k)*21 + (3-7k)*15.

Nun ist 33, die rechte Seite der Gleichung, ein Vielfaches des GGT (andernfalls wäre die Gleichung ganzzahlig nicht lösbar). Multiplikation mit 11 ergibt die Lösung :

33 = (-22+5n)*21 + (33-7n)*15 ,n beliebig ganz.

Beispiel 3 Bestimmen Sie eine ganzzahlige Lösung der Gleichung 3108473*x + 2810451*y = 1.

Eine Lösung : x = -1152247 , y = 1274432

Dezimalbruchentwicklung

Jede rationale Zahl (d.i. Quotient zweier ganzer Zahlen) läßt sich als Dezimalbruch schreiben.Dazu sind Zähler und Nenner durcheinander zu dividieren.

1. Division mit beliebig vielen Nachkommastellen

```
10 CLS:PRINT"Beliebigstellige Division a/b":PRINT
20 INPUT "a=";a:INPUT "b=";b
30 r=a:PRINT a"/"b"= ";
40 q=0
50 IF r>=b THEN q=q+1:r=r-b:GOTO 50
60 q$=STR$(q):PRINT RIGHT$(q$,LEN(q$)-1);
70 IF NOT k THEN PRINT".";:k=-1
80 r=r*10
90 GOTO 40
```

Beispiel 1

82 / 25 = 3.28000000000000000000000000000000.....

Beispiel 2

23 / 7 = 3.285714285714285714285714285714.....

Beispiel 3

13 / 12 = 1.083333333333333333333333333333333333.....

Man schreibt : 82/25 = 3,28 'abbrechende Dezimalzahl'

23/7 = $3,\overline{285714}$ 'reinperiodiche D.'

13/12 = $1,08\overline{3}$ 'gemischtperiodische D.'

Die Ziffernfolge '08' des letzten Beispiels heißt Vorperiode.

2. Dezimalbruchentwicklung

Eine rationale Zahl wird in einen Dezimalbruch umgewandelt. Dabei werden ermittelt :

- der ganzzahlige Anteil
- die Vorperiode mit Stellenzahl
- die Periode mit Stellenzahl

```
10 CLS:PRINT"Umwandlung Bruch in Dezimalzahl":PRINT
20 INPUT "Zaehler :";z:INPUT "Nenner  :";n:PRINT
30 k=0:m=n:y=z
40 FOR p=2 TO 5 STEP 3
50 e=0
60 IF INT(m/p)<>m/p THEN 90
70 IF INT(y/p)=y/p THEN m=m/p:y=y/p:GOTO 60
80 m=m/p:e=e+1:GOTO 60
90 IF e>k THEN k=e
100 NEXT p
110 g$=". ist der ganzzahlige Anteil"
120 v$="  Vorperiode der Laenge "
130 a=INT(z/n):PRINT a;g$:z=z-a*n:i=0
140 IF i=k THEN r=z:IF k>0 THEN PRINT v$;k
150 z=z*10:zi=INT(z/n):z=z-zi*n:PRINT STR$(zi);
160 IF z=0 THEN PRINT" ist abbrechende Dezimalzahl.":END
170 IF z<>r THEN i=i+1:GOTO 140
180 PRINT:PRINT" Periode der Laenge ";i-k+1:PRINT:END
```

Beispiel 1

$93/52 = 1,78\overline{846153}$

Beispiel 2

$1/1986 = 0,\overline{000503524.....30715}$

(Periodenlänge 110)

Anregung zu experimenteller Mathematik

Klären Sie folgende Fragen :

1) Wovon hägt es ab, ob ein Bruch eine abbrechende,rein periodische bzw. gemischt periodische Dezimalbruchentwicklung besitzt ?
2) Wovon hängen Perioden- bzw. Vorperiodenlängen ab ?
3) Wie sehen die Perioden in Abhängigkeit vom Zähler aus ? Testen Sie : 1/7,2/7,...6/7 sowie 1/13,2/13,...12/13.

g - adische Zahlen

Es gibt verschiedene Zahlensysteme. Neben dem vertrauten Dezimalsystem benutzt man z.B. das binäre oder auch das hexadezimale Zahlensystem.

Im Dezimalsystem (g=10) bedeutet

$43{,}75 = 4*10^{1} + 3*10^{0} + 7*10^{-1} + 5*10^{-2}$

Im binären System (g=2) bedeutet

$1011{,}01 = 1*2^{3} + 0*2^{2} + 1*2^{1} + 1*2^{0} + 0*2^{-1} + 1*2^{-2}$

das ist die Zahl 8+2+1+0,5+0,25 = 11,75 im Dezimalsystem.

Ist g größer als 10, so benötigt man Symbole für die Ziffern 10 bis g-1. Man benutzt Buchstaben, im Hexadezimalen System (g=16) A für 10 bis F für 15:

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
1 2 3 4 5 6 7 8 9  A  B  C  D  E  F  G  H  I  J  K  L  M  N

24 25 26 27 28 29 30 31 32 33 34 35
 O  P  Q  R  S  T  U  V  W  X  Y  Z
```

Im hexadezimalen System (g=16) bedeutet

$2AF{,}C = 2*16^{2} + A*16^{1} + F*16^{0} + C*16^{-1}$,

das ist die Zahl $2*256 + 10*16 + 15*1 + 12*\frac{1}{16} = 687{,}75$ im Dezimalsystem.

1 Darstellung einer g-adischen Zahl als h-adische Zahl

Folgendes Programm wandelt g-adische in h-adische Zahlen um. Die Beschränkung g,h < 37 ergibt sich aus obiger Ziffernzuordnung.

Die Ziffereneingabe kann - falls g>10 ist - in großen oder kleinen Buchstaben erfolgen. Das Ergebnis wird wie üblich mit großen Buchstaben ausgegeben (siehe Bsp. 3).

```
10 CLS:PRINT"          Umwandlung:"
20 PRINT:PRINT
30 PRINT"g-adische in h-adische Zahlen"
40 PRINT:PRINT"          1 < g,h <37"
```

```
50 PRINT:PRINT
60 PRINT:INPUT "g=";g
70 PRINT:INPUT "h=";h
80 PRINT:PRINT
90 PRINT"y(";g;")=";:INPUT y$:PRINT
100 REM **** g-adisch in 10-adisch ****
110 x=0:k=0
120 r=ASC(RIGHT$(y$,1))-48
130 IF r>48 THEN  LET r=r-32
140 IF r>9 THEN LET r=r-7
150 IF r>g-1 THEN PRINT"Ziffern zu gross fuer g":GOTO 60
160 IF r<0 THEN LET x=x/g↑k:k=0:GOTO 180
170 x=x+r*g↑k:k=k+1
180 l=LEN(y$):IF l=1 THEN 210
190 y$=LEFT$(y$,l-1):GOTO 120
200 REM **** 10-adisch in h-adisch ****
210 k=0:y$="":z$=""
220 IF x>1E+09 THEN PRINT"Rundungsfehler moeglich !":PRINT
230 b=INT(x):IF x=b THEN 310
240 z$=".":x=x-b
250 k=k+1:u=INT(h*x):r=u
260 IF r>9 THEN LET r=r+7
270 z$=z$+CHR$(r+48)
280 x=h*x-u
290 IF (k>10 OR x=0) THEN LET x=b:GOTO 310
300 GOTO 250
310 u=INT(x/h)
320 r=x-h*u
330 IF r>9 THEN LET r=r+7
340 y$=CHR$(r+48)+y$
350 IF u>0 THEN LET x=u:GOTO 310
360 REM ***** Ausgabe *****
370 PRINT"y(";h;")=";y$+z$
380 PRINT:PRINT
390 GOTO 60:END
```

Beispiel 1:

(a) Die Dezimalzahl 43,75 ist als Binärzahl darzustellen
Es ist also g=10, h=2 und y(10) = 43.75
Wir erhalten: y(2) = 101011.11

(b) Die Binärzahl 101011,11 ist als Hexadezimalzahl darzustellen:
g = 2, h = 16, y(2) = 101011.11 ergibt y(16) = 2B.C

(c) Die Hexadezimalzahl 2B,C ist in die entsprechende Dezimalzahl umzuwandeln:
g = 16, h = 10, y(16) = 2B.C ergibt y(10) = 43.75

Beispiel 2:

(a) g = 16, h = 2, y(16) = 111 ergibt y(2) = 100010001

(b) g = 25, h = 5, y(25) = 333.33333 ergibt
y(5) = 30303.03030303030

Beispiel 3:

g = 35, h = 16, y(35) = GeNug ergibt
y(16) = 177F803

Probieren wir es umgekehrt:

g = 16, h = 35, y(16) = 177F803 ergibt tatsächlich
y(35) = GENUG

Diese Wortspiele funktionieren natürlich nicht immer, da bei zu langen Wörtern und zu großen g und h die Zahlen zu groß werden und Rechner— bedingt Rundungsfehler auftreten.

Natürlich kann vorstehendes Programm noch abgemagert werden. Das folgende Programm ist für den Sharp gedacht:

2 g - adische Zahlen (Kurzprogramm)

Dieses Programm ist das gleiche wie das vorige, nur sind "überflüssige" Zeilen weggelassen:

```
10 INPUT "y=";y$
20 INPUT "g=";g
30 INPUT "h=";h
40 x=0:k=0
50 r=ASC(RIGHT$(y$,1))-48
60 IF r>48 THEN LET r=r-32
70 IF r>9 THEN LET r=r-7
80 IF r>g-1 THEN PRINT"Ziffern zu gross":GOTO 10
90 IF r<0 THEN LET x=x/g↑k:k=0:GOTO 110
100 x=x+r*g↑k:k=k+1
110 l=LEN(y$):IF l=1 THEN 130
120 y$=LEFT$(y$,l-1):GOTO 50
130 k=0:y$="":z$=""
140 b=INT(x):IF x=b THEN 220
150 z$=".":x=x-b
160 k=k+1:u=INT(h*x):r=u
170 IF r>9 THEN LET r=r+7
180 z$=z$+CHR$(r+48)
190 x=h*x-u
200 IF (k>10 OR x=0) THEN LET x=b:GOTO 220
210 GOTO 160
220 u=INT(x/h):r=x-h*u
230 IF r>9 THEN LET r=r+7
240 y$=CHR$(r+48)+y$
250 IF u>0 THEN LET x=u:GOTO 220
260 PRINT"y=";y$+z$
270 END
```

Beispiel 4: (siehe Bsp. 1)

y = 2B.C
g = 16
h = 2 ergibt y = 101011.11

Probieren Sie mal: y = 59118, g=10, h=40. Was bedeutet das?

Primzahlen

1.Primzahltest

Eine natürliche Zahl n wird (durch Probedivision) daraufhin untersucht, ob sie prim ist. Ist das nicht der Fall, wird ein Teiler angegeben.Für Zahlen $n<10^7$ ist der Zeitaufwand beim SCHNEIDER CPC max. 20s.Für die Jagd nach großen Primzahlen ist das Programm natürlich nicht geeignet.

```
10 CLS:PRINT"Primzahltest":PRINT
20 INPUT "x";x:w=SQR(x):IF x<=1 THEN 20
30 IF (x=2 OR x=3) THEN 60
40 t=2:GOSUB 80
50 FOR t=3 TO w+1 STEP 2:GOSUB 80:NEXT t
60 PRINT x;"ist prim"
70 END
80 q=x/t:IF q=INT(q) THEN PRINT t;"ist Teiler":GOTO 70
90 RETURN
```

Beispiel 1 x = 9 999 997
7 ist ein Teiler

Beispiel 2 x = 9 999 991
x ist prim.

2. Primfaktorzerlegung

Eine natürliche Zahl n wird vollständig in Primfaktoren zerlegt.

```
10 CLS:PRINT"Zerlegung in Primfaktoren":PRINT
20 INPUT "x=";x
30 IF INT(x)<x OR x=1 THEN 20
40 PRINT
50 '
60 IF x=2 THEN 230
70 t=2:z=x:GOSUB 160:IF k=0 THEN 90
80 GOSUB 190:IF z=1 THEN 240
90 t=1
100 w=SQR(z):t=t+2:IF t>w THEN IF l=0 THEN 230
110 IF t>w THEN PRINT z:GOTO 240
```

```
120 GOSUB 160:IF k=0 THEN 100
130 GOSUB 190:IF z=1 THEN 240
140 GOTO 100
150 '
160 k=0
170 q=z/t:IF INT(q)<q THEN RETURN
180 k=k+1:z=q:GOTO 170
190 l=l+1:IF k=1 THEN PRINT t:RETURN
200 PRINT t"↑"k
210 RETURN
220 '
230 PRINT x" ist prim"
240 PRINT:END
```

Beispiel 1 x = 9 999 999

$x = 3^2 \cdot 239 \cdot 4649$

Beispiel 2 x = 462841 ist prim

3. Primzahlsieb

Nach Eingabe einer Obergrenze x werden die Primzahlen $\leq x$ ausgegeben und ihre Anzahl wird ermittelt.Die maximale Obergrenze ist abhängig vom Speicher des Rechners : Durch z(k)=1 werden nämlich die Nicht-Primzahlen $\leq x$ 'markiert' und das kostet natürlich Speicherplatz.

```
10 CLS:PRINT"Primzahlsieb":PRINT
20 DEFINT z
30 INPUT "x";x:DIM z(x)
40 a=1:w=SQR(x):PRINT"2 ";:FOR k=3 TO x STEP 2
50 IF z(k)=0 THEN a=a+1:PRINT k;:IF k<w THEN GOSUB 80
60 NEXT k:PRINT
70 PRINT"Anzahl :"a:END
80 FOR v=k*k TO x STEP 2*k:z(v)=1:NEXT :RETURN
```

Beispiel : Eingabe x = 5000 als Obergrenze

Ausgabe : 2,3,5,7,11,.....,4987,4993,4999

Anzahl : 669

Rechnen mit großen natürlichen Zahlen

Diese Programme betreffen das Rechnen mit natürlichen Zahlen unter Berücksichtigung ihrer vollen Stellenzahl.
Die Programme übernehmen die zu verarbeitenden Zahlen als Strings und wandeln sie zur Weiterbearbeitung in Felder um.

1. Multiplikation

```
10 DEFINT i-s
20 CLS:PRINT"Multiplikation a*b"
30 CLEAR:PRINT:INPUT "a=";b$:INPUT "b=";c$
40 DIM k(202),m(2,202),q(2),a$(2)
50 a$(1)=b$:a$(2)=c$:FOR i=1 TO 2:l=LEN(a$(i))
60 m(i,j)=VAL(RIGHT$(a$(i),2)):l=l-2
70 IF l<1 THEN 90
80 a$(i)=LEFT$(a$(i),l):j=j+1:GOTO 60
90 q(i)=j:j=0:NEXT i
100 j=q(2):IF q(1)+j<201 THEN 140
110 PRINT"Das Ergebnis ist >10↑400":GOTO 20
120 '
130 ' Multiplikation
140 FOR l=0 TO q(1):r=0:FOR i=0 TO j+1
150 s=m(2,i)*m(1,l)+k(i+l)+r
160 k(i+l)=s MOD 100:r=INT(s/100):NEXT i:NEXT l
170 j=j+q(1):IF k(j+1)>0 THEN j=j+1
180 '
190 ' Ausgabe
200 PRINT"a*b =";STR$(k(j));:FOR i=j-1 TO 0 STEP-1
210 IF k(i)>9 THEN PRINT RIGHT$(STR$(k(i)),2);:GOTO 230
220 PRINT"0";RIGHT$(STR$(k(i)),1);
230 NEXT i:PRINT:GOTO 30
```

Beispiel : a = 145743 b = 762377

a*b = 111111111111

Beispiel : a = 12345678910111213141516171819 20

b = 12345678987654321

a*b = 152415788708887102884437338842949006211310763 20

2. Division mit Rest

```
10 CLS:PRINT "Division a/b":PRINT
20 DEFINT i-z
30 CLEAR:DIM q(202),x(202),y(202)
40 DIM z(202),m(2,202),a$(2)
50 INPUT "a=";b$:INPUT "b=";c$:PRINT
60 a$(1)=b$:a$(2)=c$:FOR i=1 TO 2:l=LEN(a$(i))
70 m(i,j)=VAL(RIGHT$(a$(i),2)):l=l-2:IF l<1 THEN 90
80 a$(i)=LEFT$(a$(i),l):j=j+1:GOTO 70
90 m(i,202)=j:j=0:NEXT i
100 j=m(1,202):FOR i=0 TO j:x(i)=m(1,i):NEXT i
110 x(202)=m(1,202):j=m(2,202):FOR i=0 TO j
120 y(i)=m(2,i):z(i)=y(i):NEXT i
130 y(202)=m(2,202):z(202)=y(202):l=y(202):q(202)=-1
140 PRINT b$" / "c$
150 IF x(202)-y(202)>=0 THEN 170
160 q(202)=0:q(0)=0:GOSUB 490:GOTO 30
170 k=x(202):n=k-l:IF n<0 THEN GOSUB 490:GOTO 30
180 IF l>0 THEN T=INT((100*X(K)+X(K-1))/(100*y(l)+y(l-1)))
190 IF l=0 THEN t=INT(x(k)/y(l))
200 IF t<>0 OR n=0 THEN GOTO 240
210 n=n-1:IF l=0 THEN t=INT((100*X(K)+X(K-1))/y(0))
220 IF l<=0 THEN 240
230 t=INT((10000*x(k)+100*x(k-1)+x(k-2))/(100*y(l)+y(l-1)))
240 GOSUB 440:GOSUB 320
250 IF v>0 THEN GOSUB 380:q(n)=t:IF q(202)<0 THEN q(202)=n
260 IF v=0 AND t>1 THEN t=t-1:GOTO 240
270 IF v=0 AND t=1 THEN t=0:GOTO 210
280 IF n<=0 THEN GOSUB 490 ELSE GOTO 170
290 GOTO 30
300 ' Vergleich von x mit z*100^n
310 ' v=1+sign(x-z*100^n)
320 j=k+1:IF k<n+z(202) THEN v=0:RETURN
330 j=j-1:IF x(j)<z(j-n) THEN v=0:RETURN
340 IF x(j)>z(j-n) THEN v=2:RETURN
350 IF j>n THEN GOTO 330 ELSE v=1
```

```
360 RETURN
370 ' x = x - z+100^n
380 j=x(202):r=0:FOR i=n TO j:x(i)=x(i)-r-z(i-n)
390 r=0:IF x(i)<0 THEN r=1:x(i)=x(i)+100
400 NEXT i:j=j+1
410 j=j-1:IF x(j)=0 AND j>0 THEN 410
420 x(202)=j:RETURN
430 ' z = t * y
440 j=y(202):r=0:FOR i=0 TO j:s=y(i)*t+r
450 z(i)=s MOD 100:r=INT(s/100):NEXT i:z(j+1)=r
460 IF r>0 THEN  j=j+1
470 z(202)=j:RETURN
480 ' Ausgabe
490 j=q(202):PRINT" = ";STR$(q(j));
500 FOR i=j-1 TO 0 STEP -1
510 IF q(i)>9 THEN PRINT RIGHT$(STR$(q(i)),2);
520 IF q(i)<=9 THEN PRINT "0"RIGHT$(STR$(q(i)),1);
530 NEXT i:PRINT:j=x(202):PRINT"Rest";STR$(x(j));
540 FOR i=j-1 TO 0 STEP -1
550 IF x(i)>9 THEN PRINT RIGHT$(STR$(x(i)),2);
560 IF x(i)<=9 THEN PRINT "0"RIGHT$(STR$(x(i)),1);
570 NEXT i:PRINT:PRINT:RETURN
```

Beispiel :

```
12345678987654321 / 111222333
 =  111000000
Rest 24654321.
```

Beispiel :

```
111111111111 / 9901
 =  11222211
Rest 0
```

3. Potenzen

```
10 CLS:PRINT"Potenzen b↑k"
20 CLEAR:DEFINT b-s
30 DIM b(202),c(202),d(202)
40 PRINT:INPUT "b=";b$:INPUT "k=";q
50 p=0:c$=b$:l=LEN(b$):n=q
60 d(p)=VAL(RIGHT$(c$,2)):l=l-2:IF l<1 THEN 80
```

```
70 c$=LEFT$(c$,l):p=p+1:GOTO 60
80 b(0)=1
90 m=INT(n/2):IF n MOD 2=0 THEN 150
100 t=j+p+1:FOR l=0 TO p:r=0
110 FOR i=0 TO j+1:s=b(i)*d(l)+c(i+l)+r
120 c(i+l)=s MOD 100
130 r=INT(s/100):NEXT i:NEXT l:IF c(t)=0 THEN t=t-1
140 j=t:FOR i=0 TO j:b(i)=c(i):c(i)=0:NEXT i
150 IF n<=1 THEN 240
160 t=2*p+1:FOR l=0 TO p:r=0
170 FOR i=0 TO p+1
180 s=d(l)*d(i)+c(i+l)+r:c(i+l)=s MOD 100
190 r=INT(s/100):NEXT i:NEXT l:p=2*p+1:IF c(t)=0 THEN t=t-1
200 p=t:FOR i=0 TO p:d(i)=c(i):c(i)=0:NEXT i
210 n=m:GOTO 90
220 '
230 ' Ausgabe
240 PRINT:PRINT"b↑k=";STR$(b(j));:FOR i=j-1 TO 0 STEP-1
250 IF b(i)>9 THEN PRINT RIGHT$(STR$(b(i)),2);:GOTO 270
260 PRINT"0";RIGHT$(STR$(b(i)),1);
270 NEXT i:PRINT:GOTO 20
```

Beispiel 1

2^{127} = 170141183460469231731687303715884105728

(Bis zum Jahr 1951 war $2^{127} - 1$ die größte bekannte Primzahl).

Beispiel 2

101010101^3 = 1030610152125272725211510060301

4. Fakultät

```
10 DEFINT i-l:DIM k(200)
20 CLS:PRINT"Fakultaet  (n<200)"
30 PRINT:INPUT "n=";n:PRINT
40 k(0)=1:j=0
50 '
60 FOR l=2 TO n:ir=0:it=0
70 FOR i=0 TO j
80 is=k(i)*l+ir:k(i)=is MOD 100
90 ir=it+INT(is/100):it=INT(ir/100)
```

```
100 ir=ir MOD 100
110 NEXT i
120 k(j+1)=ir:k(j+2)=it
130 IF it>0 THEN j=j+2:GOTO 150
140 IF ir>0 THEN j=j+1
150 NEXT l
160 '
170 PRINT n;"! =";STR$(k(j));
180 FOR i=j-1 TO 0 STEP-1
190 IF k(i)>9 THEN PRINT RIGHT$(STR$(k(i)),2);:GOTO 210
200 PRINT"0";RIGHT$(STR$(k(i)),1);
210 NEXT i
220 FOR l=1 TO 200:k(l)=0:NEXT l
230 PRINT:GOTO 30
```

Beispiel 1

```
 20 ! = 2432902008176640000
```

Beispiel 2

```
 100 ! = 933262154439441526816992388562667004907159682643816214685929638952175999932299156089414639761565182862536979208272237582511852109168640000000000000000000000000
```

Komplexe Funktionen *

Das Programm berechnet die Funktionswerte einiger elementarer komplexer Funktionen.

```
10 CLS:PRINT"Komplexe Funktionen u+iv=f(x+iy)":PRINT
20 PRINT"Berechnet werden die Funktionswerte von"
30 PRINT:PRINT:PRINT"1 Polynom"
40 PRINT"2 e↑z"
50 PRINT"3 sin z"
60 PRINT"4 cos z"
70 PRINT"5 tan z"
80 PRINT"6 sinh z"
90 PRINT"7 cosh z"
100 PRINT"8 tanh z"
110 PRINT"9 log z"
120 PRINT "10 z↑n"
130 PRINT "11 z↑1/n"
140 PRINT"12 z↑(a+bi)
150 PRINT:PRINT"13 Programmende"
160 '
170 PRINT:INPUT"Welche Funktion";e
180 CLS:IF e=13 THEN 1030
190 '
200 IF e>1 THEN 330
210 PRINT"Polynomwert p(z)":PRINT
220 INPUT "Polynomgrad";n:PRINT
230 PRINT"Koeffizienteneingabe : Re,Im
240 FOR k=n TO 0 STEP-1
250 PRINT"p("k")=";:INPUT a(k),b(k):NEXT k
260 PRINT:INPUT "Re z,Im z";x,y
270 GOSUB 480:GOSUB 390
280 PRINT "Weiterer Wert dieses Polynoms (j/n)"
290 e$=INKEY$:IF e$="" THEN 290
300 IF e$="j" THEN 260
310 CLS:GOTO 20
320 '
```

*) gegebenenfalls PI = 4*ATN(1) definieren!

```
330 PRINT:INPUT "Re z,Im z";x,y
340 '
350 ON e GOSUB 480,550,570,600,630,680,710,740,800,840,
    880,940
360 IF e>8 THEN 30
370 GOSUB 390:GOTO 30
380 '
390 PRINT:PRINT f$x"+i*"y")="u"+i*"v:RETURN
400 '
410 r=SQR(x*x+y*y)
420 IF x=0 THEN w=SGN(y)*PI/2:RETURN
430 w=ATN(y/x):IF x>0 THEN RETURN
440 IF y=0 THEN w=w+PI:RETURN
450 w=w+SGN(y)*PI:RETURN
460 '
470 ' Polynome
480 f$="p(":FOR k=n TO 0 STEP-1
490 u=r(k+1):v=i(k+1):GOSUB 530
500 r(k)=a(k)+s:i(k)=b(k)+t:NEXT k
510 r(n+1)=0:i(n+1)=0
520 u=r(0):v=i(0):RETURN
530 s=u*x-v*y:t=x*v+u*y:RETURN
540 ' e↑z
550 f$="e↑(":t=EXP(x):u=t*COS(y):v=t*SIN(y):RETURN
560 ' sin z
570 f$="sin(":t=EXP(y):u=SIN(x)*(t+1/t)/2
580 v=COS(x)*(t-1/t)/2:RETURN
590 ' cos z
600 f$="cos(":t=EXP(y):u=COS(x)*(t+1/t)/2
610 v=SIN(x)*(1/t-t)/2:RETURN
620 ' tan z
630 f$="tan(":t=EXP(y):u1=SIN(x)*(t+1/t)/2
640 v1=COS(x)*(t-1/t)/2:u2=COS(x)*(t+1/t)/2
650 v2=SIN(x)*(1/t-t)/2:n=u2*u2+v2*v2
660 u=(u1*u2+v1*v2)/n:v=(u2*v1-u1*v2)/n:RETURN
670 ' sinh z  (=sin(iz)/i)
680 f$="sinh(":t=EXP(x):u=COS(y)*(t-1/t)/2
```

```
690 v=SIN(y)*(t+1/t)/2:RETURN
700 ' cosh z  (=cos(iz))
710 f$="cosh(":t=EXP(x):u=COS(y)*(t+1/t)/2
720 v=SIN(y)*(t-1/t)/2:RETURN
730 ' tanh z  (=-tan(iz)/i)
740 f$="tanh(":t=EXP(x):v1=SIN(y)*(t+1/t)/2
750 u1=COS(y)*(t-1/t)/2:u2=COS(y)*(t+1/t)/2
760 v2=SIN(y)*(t-1/t)/2:n=u2*u2+v2*v2
770 u=(u1*u2+v1*v2)/n:v=(u2*v1-u1*v2)/n
780 RETURN
790 ' log z
800 GOSUB 810:u=LOG(r):v=w:PRINT
810 PRINT "log("x"+i*"y")="u"+i*("v"+2k*pi), k ganz"
820 RETURN
830 ' z↑n
840 INPUT "n=";n:GOSUB 810
850 r=r↑n:w=w*n:u=r*COS(w):v=r*SIN(w)
860 PRINT"("x"+i*"y")↑"n"="u"+i*"v:RETURN
870 ' z↑1/n
880 INPUT "n=";n:GOSUB 810
890 r=r↑(1/n):w=w/n:a=2*PI/n
900 FOR k=0 TO n-1
910 u=r*COS(w+k*a):v=r*SIN(w+k*a)
920 PRINT"z("k")="u"+i*"v:NEXT k:RETURN
930 ' z↑(a+bi)
940 INPUT "a,b";a,b
950 GOSUB 810:u=LOG(r):v=w
960 x1=a*u-b*v:y1=a*v+b*u
970 t=EXP(x1):u=t*COS(y1):v=t*SIN(y1)
980 PRINT:PRINT"("x"+i*"y")^("a"+i*"b")=":b=-b
990 IF INT(a)=a THEN PRINT"("u"+i*"v")*EXP("b"*2k*PI)
    ,k ganz":GOTO 1010
1000 PRINT"("u"+i*"v")*EXP(2k*PI("b"+i*"a")) ,k ganz"
1010 RETURN
1020 '
1030 END
```

Beispiel : Polynomwerte

$p(z) = (1-i)z^4 - (2+i)z^3 + iz^2 - 4iz + 3$, $p(2-3i) = ?$

Eingabe : Grad 4 p(4)=1,-1 p(3)=-2,-1 p(2)= 0,1 p(1)=0,-4
p(0)=3,0 Re z,Im z = 2,-3

Ergebnis : p(2-3i) = 87 + 290i .

Beispiel : Exponentialfunktion

exp(0.75 + 2i) = -0.880982859 + 1.92498267·i .

Beispiele : Trigonometrische Funktionen

sin(2-3i) = 9.15449915 + 4.16890696·i
cos(2-3i) =-4.18962569 + 9.1092279·i
tan(2-3i) =-3.76403E-03 - 1.00323863·i

Beispiele : Hyperbelfunktionen

sinh(1+2i) =-0.489056259 + 1.40311925·i
cosh(1+2i) =-0.642148125 + 1.06860742·i
tanh(1+2i) = 1.16673626 - 0.243458201·i

Beispiele und Erklärungen zum Logarithmus

Die komplexe Zahl z habe die Polarform

$$z = |z| \cdot e^{i\varphi} \quad \text{mit} \quad -\pi < \varphi \leq \pi .$$

Dann ist der Hauptwert des Logarithmus die Zahl

$$\text{Log } z = \ln|z| + i\varphi .$$

Unter log z versteht man alle Zahlen

$$\log z = \ln|z| + i(\varphi + 2k\pi) \quad , \; k \text{ ganzzahlig.}$$

Alle log-Werte unterscheiden sich demnach vom Hauptwert um einen Summanden $2k\pi i$.

Beispiel 1

log(-5)=1.60943791 + i(3.14159265 + 2kπ),k ganz

Der Hauptwert ist Log(-5)=ln(5)+iπ .

Beispiel 2

log(2+3i) = 1.28247468 + i(0.982793723 + 2kπ),k ganz

Beispiele : Potenzfunktion z^n (n ganzzahlig)

$(1+2i)^3 = -11 - 2i$

$(0.3-0.6i)^{-2} = -1.33333333 + 1.77777778 \cdot i$

Beispiele : Wurzelfunktion $z\uparrow 1/n$ (n natürliche Zahl)

$(-8)\uparrow 1/3$ hat die Werte $z_0 = 1 + 1.73205081 \cdot i$

$z_1 = -2$

$z_2 = 1 - 1.73205081 \cdot i$

$(2+8i)\uparrow 1/4$ hat die Werte :

$z_0 = 1.60233977 + 0.551449845 \cdot i$ $\quad z_1 = -0.551449845 + 1.60234977 \cdot i$

$z_2 = -1.60234977 - 0.551449845 \cdot i$ $\quad z_3 = 0.551449845 - 1.60233977 \cdot i$

Beispiele und Erklärungen zur allg. Potenz $z\uparrow(a+bi)$

Man definiert die allg. Potenz durch

$$z^w = e^{w \cdot \log z}$$

und erhält damit eine i.allg. unendlich vieldeutige Funktion. Hat z die Polarform $z = |z| e^{i\varphi}$, $-\pi < \varphi \leq \pi$, so ist

$$z^w = e^{w \cdot (\ln |z| + i(\varphi + 2k\pi))}$$

Der Hauptwert von z^w ist $e^{w \cdot (\ln |z| + i\varphi)}$ und die weiteren Werte unterscheiden sich davon um den Faktor $e^{w \cdot 2k\pi i}$.

Beispiel 1

$$i^i = 0.207879577 * e^{-2k\pi}, \quad k \text{ ganz} \quad \text{(alles reell)}$$

Beispiel 2

$$(-3+4i)^{1+i} = (-0.423935419 - 0.344325183i) * e^{-2k\pi}, \quad k \text{ ganz}$$

Beispiel 3

$$(0.5+2.1i)^{1.1+0.6i} = (-0.369833736 + 0.977602246i) * w_k \text{ mit}$$

$$w_k = e^{2k\pi(-0.6+1.1 \cdot i)}, \quad k \text{ ganz}$$

Beispiel 4

$$(-5)^{1.3} = (-4.76299024 - 6.55569364i) * e^{2k\pi \cdot 1.3 \cdot i}, \quad k \text{ ganz}$$

Der Faktor $e^{2k\pi 1.3 \cdot i}$ nimmt nur endlich viele Werte an : $e^{0 \cdot \pi i}$, $e^{0.2 \cdot \pi i}$, $e^{0.4 \cdot \pi i}$, $e^{0.6 \cdot \pi i}$, ..., $e^{1.8 \cdot \pi i}$, so daß $(-5)^{1.3}$ außer dem Hauptwert 9 weitere Werte besitzt.

Gammafunktion

In diesem Programm wird die Gammafunktion auf dem Intervall $[1,2]$ durch ein Polynom 8.Grades approximiert.Der Fehler liegt in diesem Bereich unter $3*10^{-7}$.

$x = 0,-1,-2,-3,...$ sind Polstellen der Gammafunktion und für andere $x \in \mathbb{R}$ läßt sich die Funktionswertberechnung durch

$$\Gamma(x+1) = x \cdot \Gamma(x)$$

auf den Fall $x \in [1,2]$ zurückführen.

Für Anwender spezieller Funktionen ist zu empfehlen :
ABRAMOWITZ , Handbook of Mathematical Functions.

```
10 CLS:PRINT"Gammafunktion":PRINT
20 a(1)=-0.577191652:a(2)=0.988205891:a(3)=-0.897056937
30 a(4)=0.918206857:a(5)=-0.756704078:a(6)=0.482199394
40 a(7)=-0.193527818:a(8)=0.035868343:a(0)=1
50 '
60 INPUT "x=";x:f=1:x=x-1
70 IF x<0 AND INT(x)=x THEN PRINT"Polstelle":GOTO 60
80 IF x<-1 THEN 140
90 IF x>30 THEN PRINT"x zu gross":GOTO 60
100 IF x<=1 THEN 160
110 f=f*x:x=x-1:GOTO 100
120 '
130 IF x>1 THEN x=x-1:f=1/f:GOTO 160
140 f=f*x:x=x+1:GOTO 130
150 '
160 y=a(8):FOR k=7 TO 0 STEP-1:y=a(k)+x*y:NEXT k
170 y=f*y*1000000:z=INT(y):r=y-z
180 y=z:IF r>0.5 THEN y=z+1
190 PRINT z/1000000
200 GOTO 60
```

Beispiele

$\Gamma(0.7) = 1.295989$, $\Gamma(3.5) = 3.323351$

Funktionen in Wahrscheinlichkeitsrechnung und Statistik

Dieses Programm berechnet folgende Funktionen :

1) Fakultät n!

2) Binomialkoeffizient $\binom{a}{m}$,a reell

3) Binomialverteilung

a) $w = \binom{n}{x} \cdot p^x q^{n-x}$

w ist die Wahrscheinlichkeit dafür, daß ein Ereignis A bei n unabhängigen Ausführungen eines Experiments genau x-mal eintrifft, wenn A bei der Einzelausführung die Wahrscheinlichkeit p besitzt und q=1-p ist.

b) $v = \sum_{k \leq x} \binom{n}{k} p^k q^{n-k}$ Verteilungsfunktion , summiert werden die Wahrscheinlichkeiten für ganzzahliges k, $0 \leq k \leq x$.

4) Poissonverteilung

a) $w = \frac{m^x}{x!} \cdot e^{-m}$ (m : Mittelwert der Verteilung)

b) $v = e^{-m} \cdot \sum_{k \leq x} \frac{m^k}{k!}$

5) Normalverteilung

a) $w = \frac{1}{s\sqrt{2\pi}} e^{-\frac{1}{2} \cdot \left(\frac{x-m}{s}\right)^2}$ Wahrscheinlichkeitsdichte

s : Standardabweichung , m Mittelwert

b) $v = \frac{1}{s\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2} \cdot \left(\frac{t-m}{s}\right)^2} \cdot dt$ Verteilungsfunktion

6) Normalverteilung , Umkehrfunktion der Verteilungsfunktion v(x) :
Zu gegebener Wahrscheinlichkeit p ,$0<p<1$, wird der x Wert berechnet, für den $v(x) = p$ gilt.

```
10 CLS:PRINT"Funktionen der Statistik":PRINT
20 PRINT"Berechnet werden :":PRINT
30 PRINT"1  Fakultaet"
40 PRINT"2  Binomialkoeffizient"
50 PRINT:PRINT"Wahrscheinlichkeitsfunktion w und"
60 PRINT"Verteilungsfunktion v fuer :":PRINT
70 PRINT"3  Binomialverteilung"
80 PRINT"4  Poissonverteilung"
90 PRINT"5  Normalverteilung"
100 PRINT:PRINT"6  Umkehrfunkt. der Verteilungsfunktion"
110 PRINT"   bei der Normalverteilung"
120 PRINT:PRINT"Welche Funktion"
130 e$=INKEY$:IF e$="" THEN 130
140 PRINT:e=VAL(e$)
150 ON e GOSUB 190,290,390,540,680,680
160 e$=INKEY$:IF e$="" THEN 160
170 CLS:GOTO 20
180 '
190 CLS:PRINT"Fakultaet":PRINT
200 INPUT "n=";n
210 y=1:e=0
220 FOR k=1 TO n:y=y*k
230 IF y>10 THEN y=y/10:e=e+1:GOTO 230
240 NEXT k
250 PRINT:PRINT "n!="y"*10↑"e
260 RETURN
270 '
280 '
290 CLS:PRINT"Binomialkoeffizient (a ueber m ,a reell)":PRINT
300 INPUT "a=";a
310 INPUT "m=";m
320 y=1:e=0
330 FOR k=1 TO m:y=y*(a-m+k)/k
340 IF y>10 THEN y=y/10:e=e+1:GOTO 340
350 NEXT k
360 PRINT:PRINT "=";y"*10↑"e
```

```
370 RETURN
380 '
390 CLS:PRINT"Binomialverteilung":PRINT
400 INPUT "n=";n
410 INPUT "p=";p
420 INPUT "x=";x
430 w=(1-p)↑n:f=p/(1-p)
440 v=w
450 FOR k=0 TO x-1
460 w=(n-k)*f*w/(k+1)
470 v=v+w
480 NEXT k
490 PRINT:PRINT "w="w
500 PRINT"v="v
510 RETURN
520 '
530 ' Poissonverteilung
540 CLS:PRINT"Poissonverteilung":PRINT
550 INPUT "Mittelwert";m
560 INPUT "         x=";x
570 f=EXP(-m):w=1
580 v=w
590 FOR k=1 TO x
600 w=w*m/k
610 v=v+w
620 NEXT k
630 w=f*w:v=f*v
640 PRINT:PRINT "w="w
650 PRINT"v=";v
660 RETURN
670 '
680 CLS:PRINT"Normalverteilung":PRINT
690 INPUT "Mittelwert         ";m
700 INPUT "Standardabweichung";s
710 IF e=6 THEN 830
720 '
730 INPUT "x=                 ";x
```

```
740 z=(x-m)/s
750 w=EXP(-z↑2/2)/s/SQR(2*PI)
760 v=0:y=z:n=2:u=1
770 v=v+y:IF ABS(y)>10↑(-9) THEN y=-y*z*z*u/n:u=u+2:
    n=n+2:y=y/u:GOTO 770
780 v=0.5+v/SQR(2*PI)
790 PRINT:PRINT "w=";w
800 PRINT"v=";v
810 RETURN
820 '
830 PRINT:PRINT"Eingabe von p     ";:INPUT p:v=1
840 IF p<0.5 THEN v=-1:GOTO 860
850 p=1-p
860 t=SQR(-2*LOG(p))
870 z=2.515517+0.802853*t+0.010328*t*t
880 n=1+1.432788*t+0.189269*t*t+0.001308*t*t*t
890 x=1000*((t-z/n)*v*s+m):y=INT(x):r=x-y
900 IF r>0.5 THEN y=y+1
910 PRINT:PRINT "x="y/1000
920 RETURN
```

Beispiel 1 : $15! = 1.30767437*10^{12}$

Beispiel 2 : $\binom{49}{6} = 1.3983816*10^{7}$

Beispiel 3 : Binomialverteilung

n=8 p=0.3 x=4 Ergebnis : w=0.1361367 v=0.942032351

Beispiel 4 : Poissonverteilung

m=2 x=3 Ergebnis : w=0.180447044 v=0.857123461

Beispiel 5 : Normalverteilung

m=0.8 s=2 x=2.44 Ergebnis : w=0.142518179 v=0.793892

Beispiel 6 : Normalverteilung

m=-2 s=0.5 p=0.8 Ergebnis : -1.579

m=0.8 s=2 p=0.793892 Ergebnis : x=2.44

I N D E X

Abstand 141
-Punkt - Punkt 141
-Punkt - Gerade 141
-Punkt - Ebene 141
-Gerade - Gerade 141
-Gerade - Ebene 141
-Ebene - Ebene 141
Amplitude 49
Anfangswertproblem 97
Ausgleichskurven 42
Ausgleichspolynom 43,44

Basislösungen 84
binär 168
Binomialkoeffizient 185
Binomialverteilung 185
Bogenlänge 62
Bruchrechnung 162

Charakteristische Gleich. 84,125,128
charakteristisches Polynom 132
Cosinusschwingung 49

Determinante 123
Dezimalbruchentwicklung 166
Diagonalform 123
Differenzenquotient 14

Echt gebrochen 70
Eigenvektor 125,128,135
Eigenwert 125,128
elementare Funktionen 45,47,48,65
euklidischer Algorithmus 165
Exponentialfunktion 179,182

Fakultät 177,185
Fläche 62

Gammafunktion 184
Gauß
-numerische Integration 62,64,65,67
-Eliminationsverfahren 104
g-adisch 168
Gebietsintegral 67
ggT
-zweier Polynome 28
-zweier Zahlen 165

Hauptachsentransformation 153
hexadezimal 168
Hilbert-Matrix 109
Hornerschema reell, komplex 33
Hyperbelfunktionen 45,48,65,179

IBM-kompatibel 58
Integration 62,64,65
Interpolation 35
Interpolationspolynom 36,132

Kegelschnitte 153,155,160
Kettenrechnung 27
Koeffizientenmatrix,Rang 104
komplexe Funktionen 179
komplexe Zahlen 147,148
Kreisfrequenz 49
Kurven 62

Lineare Dgln mit konst. Koeffizient.
-homogene 84
-inhomogene 91
-spezielle Lösung 87
lineare Gleichungssysteme 104
-eindeutige Lösbarkeit 108
-mit komplexen Koeffizienten 111
-Pivot-Verfahren 109
Linearfaktor (Polynom) 38,70
Lipschitzkonstante 99
Logarithmusfunktion 45,48,65,179

Matrizen
-Gleichung 114
-Invertierung 120
-Potenz 118
-Produkt 117
Methode der kleinsten Quadrate 42
Mises 137
Mittelwert 185

Näherungsverfahren 9
natürliche Zahlen
-Division mit Rest 175
-Fakultät 177
-Multiplikation 174
-Potenz 176
Newton 9
nichtlineare Dgln 97
Normalverteilung 185
Nullstellen 9,70
-eines Polynoms 38
-Halbierungsverfahren 15
-mehrfache 18
-mit Vorzeichenwechsel 15

Partialbruchzerlegung 70
PBZ 70
Periode 49
Phase 49

Pivot-Verfahren 109
Poissonverteilung 185
Polynome
-Division mit Rest 27,32,70
-Multiplizieren 27
-Potenzieren 27
-Zerlegen in unzerlegb. Fakt. 38,70
Primzahlen 172
Primfaktorzerlegung 172
Primzahlsieb 173
Primzahltest 172

Quadratische Gleichung, komplex 152
quadratischer Faktor, Polynom 38,70

Rationale Funktionen 70
Regressionsgerade 42
Regressionspolynom 43
Resonanz 87
Runge-Kutta 97

Schwingungen 49,54,58
-charakteristische Größen 49
Simpson-Verfahren 63
Sinusschwingung 49
spezielle Lösung
-einer linearen Dgl 91
-eines linearen Gleichungssystems 104
Standardabweichung 185
Stapel (Register) 27,148,162
Störfunktion 87,91

Ueberlagerung 50,54,58
umgekehrte polnische Notation 162

Vektorrechnung 141
Verteilungsfunktion 185
Vorzeichenwechsel 15

Wahrscheinlichkeit 185

Zerlegung (Schwingung) 50,54
zweifaches Integral 67

C. Feldmann Verlag
Am Bergfelde 28
3257 Springe 2

ISBN 3-923923-15-5